Two-dimensional curvature operators

Jan J. Koenderink

Fysisch Laboratorium, Rijksuniversiteit Utrecht, Utrecht 3584 CC, The Netherlands

Whitman Richards

Natural Computation Group, E10-120, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Received July 9, 1987; accepted February 26, 1988

Most computer vision systems use one-dimensional operators to calculate the curvature of boundaries or image contours. However, the more sophisticated biological systems use two-dimensional operators. Here we analyze how curvature can be computed by using two-dimensional operators and show the relation of this approach to the one-dimensional methods currently in use.

1. INTRODUCTION

Curvature plays an important role in the representation and recognition of two-dimensional shapes and surfaces. Although the importance of curvature extrema for shape recognition has been known for some time,[1,2] only recently were attempts made to provide a firmer theoretical foundation for such representations based on curvature.[3–8] Concurrently, several algorithms designed to extract curvature extrema were written and implemented.[9–11] To date, all of these algorithms utilize one-dimensional curvature operators: an edge list for a region or contour is extracted from the image, and this list is differentiated twice to yield curvature versus arc length. Here we show how curvature can be extracted by using two-dimensional spatial operators. In particular, our proposed mechanism returns a curvature value at a point along any isoluminance contour in a region defined by a spatial scale parameter. The choice of this scale parameter is left to the observer, who will presumably choose a fine spatial scale if textural detail along a curve is of interest or a coarse scale if the more global shape of the contour is desired.

There are several potential advantages of using two-dimensional operators rather than one-dimensional ones. Probably the most important is that two-dimensional operators recover the curvature across a range of scales of image filtering, whereas one-dimensional operators generally do not. Different results arise because, for example, smoothing the textural detail along the arc length is not equivalent to smoothing the details of this edge or contour in two dimensions (although it is shown in this paper that there is an approximate relationship between these two procedures when boundary curves or contours can be defined parametrically). A second advantage is that an area-based operator is more robust to noise than a one-dimensional operator. Finally, for those wishing to understand biological vision, the two-dimensional operator is more plausible, for it is unlikely that a primate visual system, still a parallel processor at this stage, creates an edge list of a curve that is then differentiated twice.

2. PRELIMINARIES

Let $L(x, y)$ denote a property such as luminance over some region, and let $[x(s), y(s)]$ be an image curve within this region, where s is the arc length along the curve. The curvature at a certain point s_i of the curve then depends on the derivatives of $[x(s_i), y(s_i)]$ up to the second order. However, the curvature may be expressed equally well in terms of the partial derivatives of the luminance $L(x, y)$ with respect to x and y. We adopt the following notation for these derivatives: L_x and L_y are the first derivatives of L in x and y, and L_{xx}, L_{xy}, L_{yy}, ... are the second partials in x and y, etc. These partials have important implications for the construction of our curvature operator.

In order to describe curvature at a range of spatial scales of resolution, let $L^*(x, y)$ be a blurred version of $L(x, y)$. We show here that the partial derivatives of L^* with respect to x and y are equal to the convolution of $L(x, y)$ with a certain kernel that depends on the order (in x and y) of the derivative. The implication of this construction is that we can have a spatially extended curvature operator that when applied to $L(x, y)$, yields the exact partial derivatives of $L^*(x, y)$ at a point.

3. CURVATURE AT BLOB BOUNDARIES

Assume that a region of an image has been isolated and that its boundary is defined by thresholding at some contrast level L_0. Thus, for a luminance $L(x, y)$ within the region, the equation for the boundary of the region, which we call a blob, would be $L(x, y) = L_0$, where (x, y) are Cartesian coordinates in the image plane. The curvature of this boundary can be found by implicitly differentiating the defining equation twice with respect to a coordinate, e.g., x. This yields two algebraic equations for the first and second derivatives of y with respect to x, which can be solved explicitly for d^2y/dx^2. In particular, given that $L_{xy} = L_{yx}$, we obtain

$$\frac{d^2y}{dx^2} = \frac{-L_{xx}L_y{}^2 + 2L_{xy}L_xL_y - L_{yy}L_x{}^2}{L_y{}^3}. \quad (1)$$

0740-3232/88/071136-06$02.00

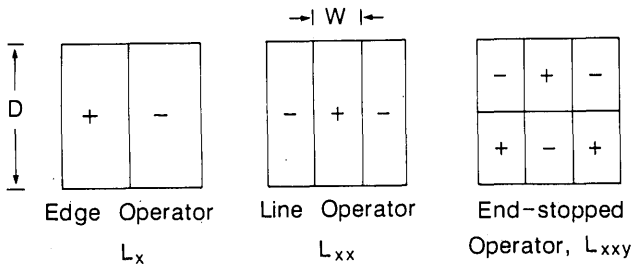

Fig. 1. Receptive-field operators used to compute curvature for blob boundaries or image contours. (See the text for a more exact description of these receptive fields.)

However, d^2y/dx^2 is the desired curvature κ in the special case in which the x axis is tangent to the blob boundary, in which case $L_x = 0$ and the solution is simply

$$\kappa = -\frac{L_{xx}}{L_y}. \tag{2}$$

Note that κ is expressed as the quotient of two partial derivatives of a function of two variables $[L(x, y)]$. Thus the numerator and the denominator can be computed by using pairs of two-dimensional (receptive-field) operators (see Fig. 1). Now we also see that this result is identical to a one-dimensional scheme for computing the curvature κ if the boundary curve is expressed parametrically as a curve $[x(s), y(s)]$, with s being the arc length.

Another potentially useful property of the method is that it yields a value of the curvature along isoluminance lines everywhere in the image (except at the stationary points $L_x^2 + L_y^2 = 0$). Thus, unlike a simple one-dimensional operator, the boundary of the blob need not be defined exactly, for a ribbon of finite width indicating the region likely to contain the boundary curve could serve as well. Such a modification of the scheme reduces the effect of discrete image sampling on the curvature computation.

4. MULTIRESOLUTION APPROACH

In practice, a slightly blurred version of the boundary may be used to describe the blob outline at several scales of resolution as well as to reduce noise. For example, as mentioned in Section 1, we may wish to zoom in on the detailed textural content of a shape, such as its ripples or hairs, while still retaining an abstract description of its overall shape. Although we cannot answer here the question of how the observer elects one scale instead of another, there is a clear choice as to how information about spatial detail should be represented. First, to represent the shape across a range of scales, it is usually preferable to blur the shape rather than its boundary and then to take the boundary of the blurred figure as the important entity.[12] Second, given that the shape rather than its boundary is to be blurred, then several authors[13–16] have shown that the most rational method is to use the diffusion equation $\Delta L = \delta L/\delta\sigma$, where σ is the resolution or scale parameter. This is identical to blurring by convolution with a Gaussian kernel. Such a scheme is the only one that yields regions and subregions across the levels of resolution (or scale space) that are well behaved in that the subregions are nested properly within their parent regions, a key property for representing the spatial structure of an object.

Because the diffusion equation is linear, the partial derivatives of a blurred image are equal to the blurred partial derivatives of the original image. By repeated partial integration of the convolution integral, one can show that this is again equal to a convolution of the original image with the corresponding partial derivatives of the Gaussian kernel.[15] Thus, returning to our curvature problem, suppose that we want the partial derivatives of $L(x, y)$ blurred with the kernel $(2\pi)^{-1}\exp[-(x^2 + y^2)/2]$. If L^* is the blurred version of L, then L_y^* denotes the convolution of L with $-y(2\pi)^{-1}\exp[-(x^2 + y^2)/2]$ and L_{xx}^* denotes the convolution of L with $(x^2 - 1)(2\pi)^{-1}\exp[-(x^2 + y^2)/2]$.

The kernels $-y(2\pi)^{-1}\exp[-(x^2 + y^2)/2]$ and $(x^2 - 1)(2\pi)^{-1}\exp[-(x^2 + y^2)/2]$ obviously may be regarded as the weights of a linear receptive-field operator applied to the original image $L(x, y)$. The latter kernel is often described as a line detector. The quotient of their outputs, by Eq. (2), is the exact boundary curvature of the blurred image, provided that the x–y directions have been chosen appropriately.

In practice, we may use a cluster of gradient detectors of various orientations with weights $\rho\cos(\phi - \phi_0)(2\pi)^{-1}\exp(-\rho^2/2)$, where ρ and ϕ are polar coordinates and ϕ takes on all orientations $(0–\pi)$. (A similar remark applies to the line detectors.) The receptive-field operator with the maximum output then indicates the orientation of the boundary and the local (x, y) coordinate frame. The process can be repeated for several size operators to set up a multiresolution hierarchy.[17] Note that such a local coordinate frame also obviates the need to introduce the mixed partial L_{xy} to perform coordinate rotation.

5. CURVATURE ALONG IMAGE CONTOURS

An image contour, unlike the isoluminance contours associated with blobs, may be viewed as a line having a certain width and contrast. The curvature computation for such linelike image contours differs somewhat from that presented earlier for blob boundaries.

Let us model an image contour as

$$L(x, y) = L_0 + ax + \tfrac{1}{2}bx^2 + \tfrac{1}{2}cy^2 + \text{higher-order terms}, \tag{3}$$

where $\|b\| \ll \|c\|$; i.e., in a local coordinate frame, the line runs in the x direction. With this model, $L_y = 0$ for $y = 0$ (thus the x axis appears as a ridge), and the luminance changes rapidly in the y direction. Dark and light lines are distinguished by negative and positive values, respectively, of the parameter c; the magnitude of $\|c\|$ is a measure of the width of the line. For lines defined by luminance contrast, the constant and linear terms of Eq. (3) are more or less irrelevant.

The higher-order terms of Eq. (3) modulate the image contour in different ways: the third-order terms describe the contour curvature, as can be seen by adding the general cubic,

$$\tfrac{1}{6}(px^3 + 3qx^2y + 3rxy^2 + sy^3), \tag{4}$$

and then completing the square terms of $L(x + \xi, \eta)$. The resulting expression is

$$L(x+\xi,\eta) = \tfrac{1}{2}(b+px)\xi^2 + qx(\xi\eta) + \tfrac{1}{2}(c+rx)\eta^2 + \text{res}, \tag{5}$$

where res contains a constant plus first-, third-, and higher-order terms. Except for the origin ($x = 0$), there now appears a mixed term ($\xi\eta$), which indicates that the direction of the line has changed. We may find this change by eliminating the mixed term by an infinitesimal rotation of our local coordinates,

$$\xi = \cos\phi\,\xi' + \sin\phi\,\eta', \tag{6a}$$

$$\eta = -\sin\phi\,\xi' + \cos\phi\,\eta', \tag{6b}$$

to obtain the following relation to the first-order ϕ and x:

$$L(x+\xi',\eta') = \tfrac{1}{2}(b+px)\xi'^2 + [qx + \phi(b-c)]\xi'\eta' + \tfrac{1}{2}(c+rx)\eta'^2 + \text{res}'. \tag{7}$$

The mixed term ($\xi'\eta'$) can now be removed by a rotation over

$$\phi = \frac{q}{c-b}\,x. \tag{8}$$

Thus, when the contour direction turns by ϕ for a distance x along the line, then the curvature of the line becomes

$$\kappa = \frac{\delta\phi}{\delta x} = \frac{q}{c-b}. \tag{9}$$

In terms of the derivatives of L, this is

$$\kappa = \frac{L_{xxy}}{L_{yy} - L_{xx}} \approx \frac{L_{xxy}}{L_{yy}}, \tag{10}$$

where the approximation is valid when $\|L_{yy}\| > \|L_{xx}\|$, which is the condition for an image contour to look linelike.

In terms of a blurred image, the result [relation (10)] means that we first use line receptive-field operators to determine the x direction {the partial derivative L_{yy} is the output of an operator with weight $(y^2-1)(2\pi)^{-1}\exp[-(x^2+y^2)/2]$}. Second, we find the output of a receptive-field operator with weight $-y(x^2-1)(2\pi)^{-1}\exp[-(x^2+y^2)/2]$ to determine the value of the mixed partial derivative L_{xxy}. (This receptive-field type corresponds to the classical end-stopped line detector.) The quotients of the outputs of the two operators are then the curvature of the line. Again, the result of this two-dimensional operation is identical to that obtained by a one-dimensional approach for a parametrically defined contour.

6. USING LINE OPERATORS AT EDGES

Line operators differ from edge operators simply by an extra order of differentiation in the direction perpendicular to the edge. Thus it should not be surprising that relation (10) for computing curvature on a contour should resemble Eq. (2) for computing curvature along an edge, with the difference being an extra partial in y for contour curvature. We might ask, therefore, if line operators of the type L_{xxy} can be used to estimate curvature at edges. We present two schemes. The first is a linear solution that entails adding a higher-order term to the line operator; the second capitalizes on nonlinearities in the visual system.

A. Linear Solution

If we position a line operator symmetrically about a step edge, there is no response for a linear system. Rather, the maximum response occurs when the operator's mask lies slightly off the edge. However, rather than displacing the mask, we wish to modify the form of the operator so that its response to the edge is given directly at any point on the edge profile. This is equivalent to estimating L_{xxy} at a point $(x, y + \delta y)$, where y is perpendicular to the edge (or isoluminance contour). This estimate in the y direction is determined by the rate of change of L_{xxy}, namely, L_{xxyy}. Accordingly, the modified L_{xxy} operator at point (x, y) takes the form

$$L_{xxy}^{*} = L_{xxy} + \tfrac{1}{2}\delta y L_{xxyy}. \tag{11}$$

Such an operator has the obvious disadvantage of requiring fourth-order partials, which cannot be expected to be recovered reliably. Although these higher derivatives could be avoided by comparing the outputs of the third-order operators within a neighborhood (rather than at a point as above), we suggest in Subsection 6.B a simpler nonlinear scheme that has some biological plausibility.

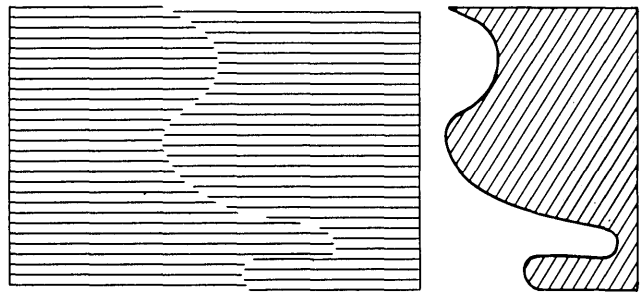

Fig. 2. Curvature along an edge created by displacing a texture (left) is much more difficult to extract than is curvature along a solid edge or an image contour (right).

B. Nonlinear Solution

Biological systems are notoriously nonlinear. Many illusions of contrast at edges result from this fact.[18–20] If an early step in visual processing entails Gaussian filtering combined with a Laplacian, then the clipped Laplacian of the edge of a blob yields a linelike contour.[21] Our linear line solution (10) can then be applied directly to this enhanced edge. This scheme predicts that curvature cannot be seen easily along edges that do not exhibit a nonlinearity across the edge. Figure 2 illustrates one such edge.

7. OPTIMUM GEOMETRIES AND SAMPLING

A. Aspect Ratio

Referring to Fig. 1, it is obvious that, for image contours of width W, the line operator L_{yy} that gives the maximum output also has a center width W. Likewise, the stopped-end operator L_{xxy} must be matched appropriately to the width of the image contour. However, here we have a problem. For low curvature an optimal receptive field of the type L_{xxy} would have a small width-to-length ratio, as illustrated in the lower panel of Fig. 3, whereas for high curvature the more appropriate L_{xxy} field would be roughly as wide as it is long. For low curvature the length-to-width ratio of the field may be as high as 6:1, whereas for high curvature this ratio reaches a lower limit of 1.5:1 for circular arcs and falls below 1:1 for highly acute vertices. Reducing the aspect

ratio to 1.5:1 or 1:1 also has the disadvantage of impairing orientation discrimination.

If we propose that the same receptive fields that discriminate orientations are used also in the curvature computation, then the aspect ratio is roughly 6:1. This estimate follows from three different derivations. First, the angular half-width[22] of orientation tuning is roughly 20 deg. This implies that a stimulus rotation of ±10° about the center of the receptive field causes the response to fall to zero. Hence the aspect ratio is about tan 10° = 1:5. The second estimate comes from Wilson's model for curvature discrimination,[23] in which, again, the appropriate aspect ratio is found to be 5:1. (Subsequently a field with an aspect ratio of 6:1 was found to handle more data.) Third, we have estimates obtained from ophthalmic migraine illusions.[24] There is little support for 1.5:1 or 1:1 aspect ratios, even in the neurophysiological literature.

B. Operator Sensitivity

Given an aspect ratio fixed at 6:1, the curvature computation is impaired at high curvatures at the expense of retaining orientation discrimination. (The advantage of not sacrificing orientation sensitivity is shown below.) To estimate the high-curvature cutoff of our curvature operator, we use a crude model proposed by Békésy,[25] which was often found to be satisfactory in other analyses. Consider the image contour depicted in Fig. 4. The gradient operator L_{yy} of center width W and length D shown in Fig. 1 delivers an optimal output signal $O_{yy} = |cWD|$, where c is the contrast $(L - B)/(L + B)$ of the contour with L and B being, respectively, the line and background luminances. Similarly, for the matched partner, the end-stopped operator L_{xxy}, the output O_{xxy} is $c\theta WD$ for small θ. Thus the calculated curvature becomes

$$K_{\text{calc}} = \frac{O_{xxy}}{O_{yy}} = \theta \tag{12}$$

(i.e, the total angular change over the receptive field) and is independent of the contrast.

In order to establish the sensitivity of the output ratio, consider the receptive field as an ideal detector of contrast. Given that contrast ranges from ±1, we can calculate the variance, provided that we know the expected distribution of contrasts. Two simple cases come to mind. First, if the distribution of contrasts is random over the interval, then the variance is 1/3. Second, if the distribution is triangular with zero mean (first approximation to a Gaussian), then the variance is 1/6. Both of these values are close to the estimate of 1/4 that is based on a theoretical analysis of reflectance functions for randomly distributed objects.[26] The expected neural signal for each operator is thus of the order $\frac{1}{2}WD$ or $\frac{1}{2}\theta WD$. Assuming that the neural signal is subject to quantum-type noise,[27–29] the standard deviation of each operator's response is equal to $(\frac{1}{2}WD)^{-1/2}$. For small θ, this yields the following tolerance for the paired operators:

$$K_{\text{meas}} = \theta \pm 2/(WD)^{1/2}. \tag{13a}$$

If the aspect ratio ϵ of a receptive field is defined as D/W and the physiological constraint of $\epsilon = 6$ is imposed, Eq. (13a) becomes approximately

$$K_{\text{meas}} = \theta \pm 1/W. \tag{13b}$$

From this result, we see that the lower limit on the accuracy of the curvature calculation depends solely on the width of the receptive field that matches the contour width. The largest values of $1/W$ occur for the smallest line masks, which have a width[30] of about 2 arcmin. The highest curvature detectable by our operator would thus be on the order of 30 deg^{-1}. This is exactly the cutoff curvature found recently by Wilson and Richards.[31]

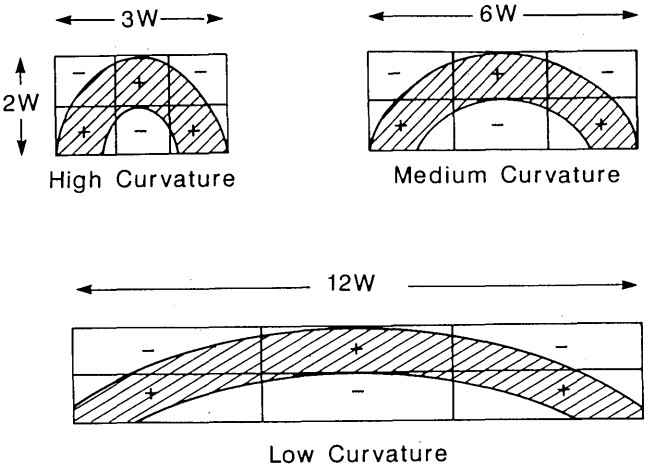

Fig. 3. For a given contour width W, the optimal aspect ratio for a receptive-field operator of type L_{xxy} depends on the curvature and the width of the contour.

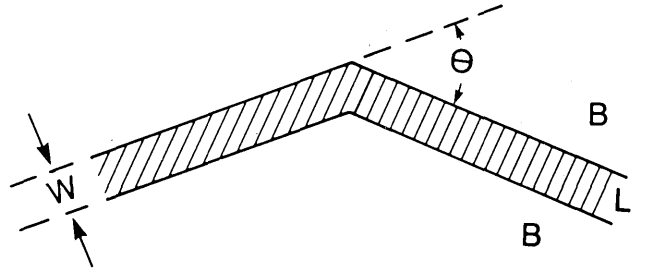

Fig. 4. Simplified model for a curved image contour of width W and luminance L against a background of luminance B.

8. COCIRCULARITY AND CONNECTION NEIGHBORHOOD

A. Point Versus Region Calculations

The preceding analysis describes curvature calculation using second- and third-order operators applied at a point in the image. An alternate scheme is to use lower-order operators that are compared at different points. The use of lower-order operators has the obvious advantage of robustness to noise and computational errors. However, the disadvantage is that comparisons between two different points requires matching local coordinate frames. Specifically, if curvature is to be computed by comparing tangent orientations at two different points on a curve, then the two coordinate frames must be registered. To bring local coordinate frames into register as one moves along a curve requires an examination of the covariant derivatives of the frame vector with respect to the particular direction of motion. The relation between two frames in the field of frames can then be described by so-called connection equations. (The

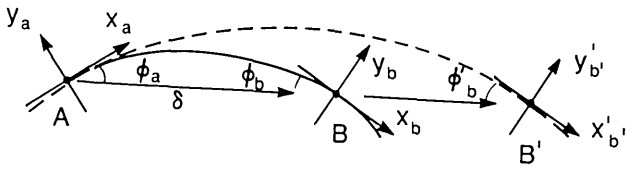

Fig. 5. Local coordinate frames at A, B, and B′ satisfy the cocircularity constraint when $\phi_a = -\phi_b$.

Frenet formulas constitute one common example.) the propagation of such local coordinate frames is well behaved, because the geodesics of the mapping do not cross. We apply the connection method to curvature by using the notion of cocircular frames.

B. Cocircularity

Consider the arc illustrated in Fig. 5, with local coordinate frames at points A and B and with the x axis set by the tangent to the contour as in the previous sections. We define two tangents, x_a and x_b, as being cocircular if and only if there is a circle to which they are both tangent.[32] Note that by this definition the tangents make equal but opposite angles ϕ to the line joining the points of tangency. Thus cocircularity defines a symmetry relation between the tangents and relates to several schemes for recovering axes to shapes.[9,33,34] In the limiting case in which the radius ρ of the circle becomes infinite, we obtain collinearity.

As we move point B along the curve away from point A, the local coordinate frame will rotate. As previously mentioned, the relation between these two frames describes the connection equations of the frame field from which curvature can be derived.[35,36] Referring to Fig. 5, we see that in the special case of cocircularity the relation between the coordinate frames is a function of two variables,

$$\phi_b = f_a(\delta, \phi_a), \tag{14}$$

namely, the orientation ϕ_a of the distance vector δ between AB to the new coordinate system. However, in the case of cocircularity, $\phi_b = -\phi_a$, so if the distance $|\delta|$ is known, then the common circle and hence the curvature are determined uniquely. We can thus define a bilocal operator for curvature by specifying an orientation ϕ and a displacement δ, where, for small angles ϕ,

$$K_{\text{bilocal}} = \frac{2\phi}{\delta}. \tag{15}$$

As discussed by Spivak,[35] this is an example of a connection operation in which the geometry at a point is extended by using lower-order derivatives plus comparisons between local coordinate frames. It seems to us that the visual system must be forced to use such bilocal operations in order to avoid unstable, higher-order derivatives such as those required by the scheme in Subsection 6.A.

C. Sensitivity

Once again, we can estimate the visual system's capability to use Eq. (15) to compute curvature. With orientation sensitivity set at 10 deg as before, and using the same 2-arcmin resolution, the maximum of K_{bilocal} is about 10 deg^{-1}, or about one third the maximum curvature obtainable with the point operator. This estimate agrees well with recent psychophysical results.[31]

9. DISCUSSION

Our proposals for calculating the curvature of the edge of a blob or an image contour differ from other earlier proposals in several respects. First, we distinguish between local and bilocal types of operators. Second, we use two-dimensional operators throughout. These operators exhibit properties of higher derivatives of the so-called simple receptive-field types.

With the exception of that of Parent and Zucker,[32] most previous schemes for computing curvature from images proposed the equivalent of differentiating a histogram of tangent orientation versus arc length along a curve.[9-11,37-39] Although the operator used to calculate the tangent is two dimensional, the next stage of differentiation is not. We have shown that for a parametric description of an image contour, the result is the same as that obtained by our local scheme. But most image boundaries are not sharp, nor are the contours of negligible width, so the convergence between the one- and two-dimensional methods does not generally occur in practice. We feel that our method is the more correct in that it can be applied to ribbons along a contour where the luminance contrast may vary. Furthermore, the one-dimensional scheme faces choosing scales of resolution for both the two-dimensional tangent operator as well as the succeeding one-dimensional curvature-differentiating operator, whereas our scheme must pick only one two-dimensional scale, which is optimally set by the contour width. [Note that this width does not necessarily fix the size of the tangent operator for the previously described (2×1)-dimensional methods.]

By direct application of high spatial derivatives of two-dimensional operators, our local method also differs from the models used by Wilson,[23] Watt,[40] and Watt and Andrews[41] to predict acuity limits for curvature. These authors assumed as the basic elements two tangent operators (i.e., a line detector L_{yy}) rotated through the minimum angle $\phi_{\min}$. This approach is bilocal and relates to connection geometry as discussed in Section 8. In particular, note that our local method assumes that the curvature is calculated from the ratio of L_{xxy} to L_{yy} rather than from $\mathrm{d}(L_{yy})/\mathrm{d}\theta$. The result is the same, namely, $\kappa_{\text{calc}} \doteq \theta$, but the method is different.

Finally, in order to avoid an explosion in receptive fields of increasingly high order, we suggest extending the local geometry with bilocal operations as formalized in differential geometry. Such computations constitute primitive grouping operations, of which cocircularity is but one simple example.

ACKNOWLEDGMENTS

This work was supported by a NATO grant for international collaboration, by U.S. Air Force Office of Scientific Research Image Understanding contract F49620-83-C-0135, and by National Science Foundation grant 8312240-IST. Hugh Wilson's comments were appreciated. Technical support was provided by William Gilson.

REFERENCES

1. K. Koffka, *Principles of Gestalt Psychology* (Harcourt, Brace, New York, 1935).
2. F. Attneave, "Some informational aspects of visual perception," Psychol. Rev. **61,** 183–193 (1954).
3. M. A. Fischler and R. C. Bolles, "Perceptual organization and curve partitioning," in *Proceedings of the 1983 Image Understanding Workshop*, L. Baumann, ed. (Science Applications, McLean, Va., 1983), pp. 224–232.
4. J. J. Koenderink and A. Van Doorn, "The shape of smooth objects and the way contours end," Perception **11,** 129–137 (1982).
5. J. J. Koenderink and A. Van Doorn, "Perceptions of solid shape and spatial layout through photometric invariants," in *Cybernetics and Systems Research*, R. Trappl, ed. (North-Holland, Amsterdam, 1982).
6. D. D. Hoffman and W. A. Richards, "Parts of recognition," Cognition **18,** 65–96 (1984).
7. H. Asada and M. Brady, "The curvature primal sketch," A.I. Memo No. 758 (Massachusetts Institute of Technology, Cambridge, Mass., 1984).
8. H. L. Resnikoff, *The Illusion of Reality: Topics in Information Science* (Springer-Verlag, New York, 1987).
9. M. Brady and H. Asada, "Smooth and local symmetries and their implementation," Int. J. Robotics **3,** 36–61 (1984).
10. B. M. Dawson and B. Treese, "Computing curvature from images," in *Applications of Digital Image Processing VII*, A. G. Tescher, ed., Proc. Soc. Photo-Opt. Instrum. Eng. **504,** 175–182 (1984).
11. W. Richards, B. Dawson, and D. Whittington, "Encoding contour shape by curvature extrema," J. Opt. Soc. Am. A **3,** 1483–1491 (1986).
12. J. J. Koenderink and A. J. Van Doorn, "Dynamic shape," Biol. Cybern. **53,** 383–396 (1986).
13. A. P. Witkin, "Scale space filtering," presented at the 7th International Joint Conference on Artificial Intelligence, Karlsruhe, Federal Republic of Germany, August 1983.
14. J. Babaud, A. Witkin, and R. Duda, "Uniqueness of the Gaussian kernel for scale-space filtering," Fairchild Tech. Rep. 645, (Fairchild Laboratory for Artificial Intelligence Research, Palo Alto, Calif., 1983).
15. J. J. Koenderink, "The structure of images," Biol. Cybern. **50,** 363–370 (1984).
16. A. L. Yuille and T. Poggio, "Fingerprint theorems for zero crossings," J. Opt. Soc. Am. A **2,** 683–692 (1985).
17. J. J. Koenderink and A. Van Doorn, "Representations of local geometry in the visual system," Biol. Cybern. **55,** 1–9 (1986).
18. E. Mach, "Über die Werkung der räumchen Vertheilung des Lichtreizer auf die Netzhaut," Sitzungsber. Math-Naturwiss. Kl. Kaiser. Akad. Wiss. Wien **52,** 303–322 (1865).
19. F. Ratliff, *Mach Bands: Quantitative Studies on Neural Networks in the Retina* (Holden-Day, San Francisco, 1965).
20. T. N. Cornsweet, *Visual Perception* (Academic, New York, 1970).
21. W. Richards, H. K. Nishihara, and B. Dawson, "CARTOON: a biologically motivated edge-detection algorithm," A.I. Memo No. 668 (Massachusetts Institute of Technology, Cambridge, Mass., 1982).
22. J. G. Daughman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," J. Opt. Soc. Am A **2,** 160–1169 (1985).
23. H. R. Wilson, "Discrimination of contour curvature: data and theory," J. Opt. Soc. Am. A **2,** 1191–1198 (1985).
24. W. Richards, "The fortification illusions of migraines," Sci. Am. **224** (5), 88–96 (1971).
25. G. von Békésy, "Neural inhibitory units of the eye and skin: quantitative description of contrast phenomena," J. Opt. Soc. Am. **50,** 1060–1070 (1960).
26. W. Richards, "Lightness scale from image intensity distributions," Appl. Opt. **21,** 2569–2582 (1982).
27. A. Rose, "The sensitivity of the human eye on an absolute scale," J. Opt. Soc. Am. **38,** 196–208 (1948).
28. M. A. Bouman, "History and present status of quantum theory in vision," in *Sensory Communication*, W. Rosenfeld, ed. (Wiley, New York, 1961), pp. 377–401.
29. P. Zuidema, "Development and present status of the quantum concept in visual psychophysics," in *Limits in Perception*, W. van der Gund and J. Koenderink, eds. (VNU Press, Utrecht, The Netherlands, 1984), pp. 5–48.
30. H. R. Wilson, "Responses of spatial mechanisms can explain hyperacuity," Vision Res. **26,** 453–469 (1986).
31. H. R. Wilson and W. Richards, "Mechanisms of curvature discrimination," J. Opt. Soc. Am. A **3**(13), P13 (1987).
32. P. Parent and S. W. Zucker, "Trace inference, curvature consistency, and curve detection," Rep. No. CIM-86-3 (McGill Research Center for Intelligent Machines, Montreal, Quebec, Canada, 1985).
33. H. Blum, "Biological shape and visual science: Part 1," J. Theor. Biol. **38,** 205–287 (1973).
34. M. Leyton, "Symmetry-curvature duality," Comput. Vision Graphics Image Process. (to be published).
35. M. Spivak, *A Comprehensive Introduction to Differential Geometry IV* (Publish or Perish, Boston, Mass., 1975).
36. N. Prakash, *Differential Geometry: An Integrated Approach* (McGraw-Hill, New York, 1981).
37. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis* (Wiley, New York, 1973).
38. D. H. Ballard and C. M. Brown, *Computer Vision* (Prentice-Hall, Englewood Cliffs, N.J., 1982).
39. B. K. P. Horn, *Robot Vision* (MIT Press, Cambridge, Mass., 1986).
40. R. J. Watt, "Further evidence concerning the analysis of curvature in human foveal vision," Vision Res. **24,** 251–253 (1984).
41. R. J. Watt and D. P. Andrews, "Contour curvature analysis: hyperacuities in the discontinuation of detailed shape," Vision Res. **22,** 444–460 (1982).