A REPRESENTATION FOR IMAGE CURVES

David H. Marimont

Artificial Intelligence Laboratory
Stanford University
Stanford, California 94305

ABSTRACT

A representation for image curves and an algorithm for its computation are introduced. The representation is designed to facilitate matching of image curves to completely specified model plane curves and estimation of their orientation in space, despite the presence of noise, variable resolution, or partial occlusion. This is an important subproblem of model-based vision. A curve may be represented at a variety of scales, and a strategy for selecting natural scales is proposed. At each scale, the representation is simply a list of positions in the plane, with tangent directions and curvatures specified at each position; each curvature is either a zero or an extremum (hereafter critical points). The algorithm for computing the representation involves smoothing with gaussians at different scales, extracting the critical points from the smoothed curves, and using dynamic programming to construct a list of critical points which best approximate the curve for each length of list possible. We propose to examine the tradeoff between the error of the approximation and length of the lists to find natural scales.

I. INTRODUCTION

In this paper we describe a representation for image curves designed to serve as input to the following computation: given a database of model plane curves, and an image containing the projection of one or more of them, decide which model curves it contains and estimate their positions and orientations in space. This is model-based vision applied to plane curves rather than to arbitrary three-dimensional objects, as in [Brooks 1981] or [Goad 1983]; even this drastic restriction is still an important problem, since the edges of three-dimensional models and their bounding contours are often plane curves.

At an abstract level, our design methodology has two phases. The first is to identify those characteristics of image curves which enable computing a desired level of reliability in model matches and viewpoint estimates at minimum cost. Next, a representation for those characteristics is selected to serve as input to a program which matches models and estimates viewpoints. Representations are judged by the extent to which they make it possible for a program, at least in theory, to achieve any specified reliability at minimum cost. These considerations lead to the following design criteria:

1. The representation must exhibit partial invariance with respect to viewpoint, so that matching can take place by comparing models to representations, rather than comparing models projected at all possible viewpoints to representations. The space of possible viewpoints is simply too large for the latter approach to be feasible computationally.
2. The representation must deal effectively with changes in the resolution of the image curve; since the curve can appear at any distance from the camera, the resolution at which it is imaged can vary widely, so that details that are clear in the model may be unavailable in the image.
3. The representation must be insensitive to noise introduced by imaging, which both obscures fine details and introduces spurious ones.
4. The representation must be robust with respect to partial occlusion of the model curve to be useful in any real application.
5. The representation must provide a range of scales of description for image curves for reasons of computational economy. Coarse descriptions can be used when error tolerances are high enough to justify eliminating irrelevant detail which needlessly overburdens the computation, while fine descriptions are available when the demand for the higher quality results they produce justifies the added computational cost.

* This report describes work done at the Stanford Artificial Intelligence Laboratory. It was supported by the Advanced Research Projects Agency of the Department of Defense under contract N00039-80-C-0250.

II. OVERVIEW OF THE REPRESENTATION

The representation described here is designed with these requirements in mind. The representation has multiple scales. At each scale, it consists of a list of points in the plane, with tangent direction and signed curvature specified at each point; each curvature is either a zero or an extremum. (We refer to such points as critical points, and following spline terminology, we call each element of these lists a knot.) The automatic selection of "natural" scales is being explored.

A curvature-based representation has attributes which help make it insensitive to changes in viewpoint. In the plane, curvature is invariant with respect to rotation and translation, and curvature ratios are invariant with respect to scale. The use of extrema and zeros of curvature provides insensitivity with respect to the projection of a plane curve oriented arbitrarily in space. A representation of an image curve based on these features will be invariant in some respects as a function of viewpoint of the model curve and deform slowly or predictably in others, thus facilitating matching of image curves to models and estimation of viewpoint.

The availability of multiple, natural scales of representation serves several purposes. It helps provide insensitivity to changes in the resolution of image curves. It provides flexibility in meeting the quality-cost tradeoff demands of a particular task. Finally, it helps discount the effect of noise, which may influence the representation at a very fine scale, but usually not at coarser ones.

Insensitivity to partial occlusion is provided by the fact that each knot in the lists of knots comprising the representation has local support, so that it contains information only about that portion of the curve between the knots adjacent to it. Thus if part of the image curve is absent, the representation of the parts which remain is not necessarily affected. The sensitivity of matching and location estimation to partial occlusion of the model curve then depends on how effectively these operations can proceed based only on a subset of the information available from an unoccluded curve.

The algorithm for computing the representation begins with a list of points in the plane, perhaps the output of an edge detector. We will refer to this list as the original sampled curve. The sampled curve is smoothed with gaussians at several different resolutions. Critical points on these smooth curves are found, and position, tangent direction, and curvature are estimated at each. These knots from different scales of smoothing are *candidates* for inclusion in the lists of knots that will ultimately represent the curve.

All the knots are considered together without regard to scale of smoothing in a graph structure which represents all possible lists of knots covering the entire sampled curve. One pass of dynamic programming is used to find each possible fixed-length list of knots which when considered as knots in a spline best approximates the original curve. That is, for each possible number of knots, that set of knots which minimizes the approximation error is chosen from the candidates. Thus smoothing at different scales produces candidate knots, while an approximation error criterion selects from them and combines them in the final representation. Those lists of knots which correspond to natural scales of representation will ultimately be selected by examining the tradeoff between the length of the lists and their approximation error.

III. EXTREMA AND ZEROS OF CURVATURE

Claims for the relevance of extrema and zeros of curvature to the perception of curves have come from both psychology and computer vision. [Attneave 1954] demonstrated experimentally the importance of curvature maxima in recognizing known objects. [Hoffman 1982] suggested segmenting curves at (signed) curvature minima, provided experimental evidence that humans did so, and implemented a program to segment curves on this basis. Others who have suggested the use of critical points include [Duda and Hart 1973], [Brady 1982], and [Hollerbach 1975].

Our claim for the relevance of critical points follows from the mathematics of the specific computational task for which the representation is to serve as input. In this section we present a few results which demonstrate why zeros and extrema of curvature provide information useful for recognizing and estimating the orientation of known plane curves in space.

Even though the image curves to be represented are *perspective* projections of plane curves in space, our analysis is based on *orthographic* projection, which for our purposes is a suitable approximation for analyzing the behavior of of curvature extrema and zeros. The basic imaging situation consists of the image plane containing the image curve and the object plane containing the object curve (a curve from the database of model plane curves). The object curve is projected onto the image plane by dropping the normal from the object point to the image plane.

The relationship between curvature in the object and curvature in the image is the heart of the analysis. While its derivation is beyond the scope of this paper, the most important consequences can be stated quite simply. First, zeros of curvature in the object curve always project to zeros of curvature in the image. (This is the differential form of the well-known fact that straight lines in space always project to straight lines in the image.)

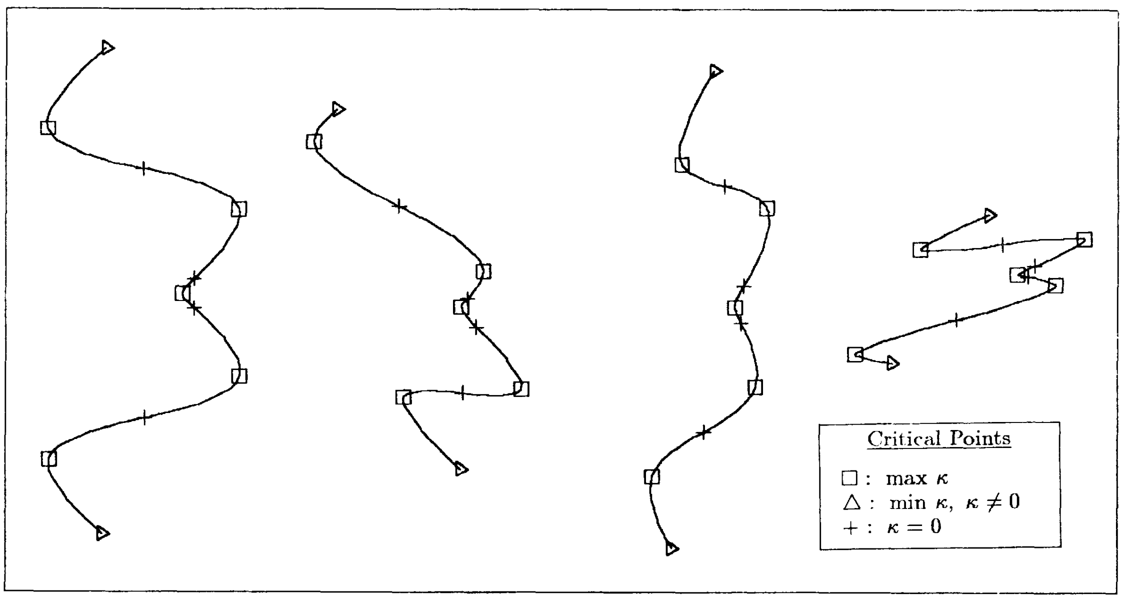

Figure 1: *The stability of critical points under orthographic projection. Left, the critical points of a plane curve. On the right, the curve is projected orthographically at various orientations and the critical points of the resulting curves are marked. The stability of their critical points aids in matching the curves to models and estimating their orientation.*

Second, as long the object plane is viewed "from above," that is, if the angle between the normals to the image and the object planes is less than π, the sign of curvature of an object point does not change under projection. If the curve is viewed "from below," the sign of curvature *always* reverses. (In the degenerate case, when the object curve is viewed "edge on," with the object plane orthogonal to the image plane, the object curve projects to a straight line and all image curvatures are zero.)

This means that the pattern of curvature sign changes along a curve is invariant under projection, except in the degenerate case. Also, since it follows that zeros of curvature are never *introduced* by the projection, except in the degenerate case, they too are invariant under projection.

The analysis of curvature extrema proceeds by differentiating the relationship between curvature in the object and curvature in the image. The interpretation of the result is more difficult and still continuing, but our preliminary conclusions are that that curvature extrema in the image move about stably and predictably as a function of viewpoint, that new ones do not appear, and old ones do not disappear, except in isolated or degenerate cases.

Furthermore, as an extremum becomes more pronounced, becoming either locally straight on the one hand or a tangent discontinuity (a cusp or corner) on the other, the more invariant under projection the location of the extremum becomes. (Here a zero of curvature is considered a minimum of *unsigned* curvature.) This is not surprising, since where a curve is locally straight, curvature is zero, which as we have seen is a projective invariant. Cusps or corners, of course, remain cusps or corners from any viewpoint, and so are projective invariants as well.

IV. MONOTONICITY OF CURVATURE

This section would be unnecessary but for an unfortunate mathematical reality: given two positions in the plane, each with a tangent direction and curvature, it is not always possible to draw a smooth path between the positions which agrees with the information at the endpoints and contains no curvature extrema. Thus, precautions must be taken when knots are assembled into lists to ensure that smooth paths monotonic in curvature can be drawn between adjacent knots. Otherwise, the representation itself implicitly introduces spurious curvature extrema.

The test for this monotonicity curvature relation between knots is quite simple. First, since there are knots at both zeros and extrema, we can narrow the problem somewhat, since paths never need be drawn between knots with curvatures of opposite sign. Consider the case when both curvatures are positive, and recall that the osculating circle at a point on a curve is that circle tangent to the curve at the point with radius equal to one over the curvature at the point, and lying to the same side of the tangent as the curve itself. Two knots define two osculating circles. It is not hard to show that to draw a monotone curvature path interpolating the knots, the larger osculating circle must completely contain the smaller, as in the leftmost subfigure of Figure 2.

This test checks the feasibility of a monotone curvature path between two knots with the same sign of curvature. When one of the two knots to be tested has zero curvature, its curvature is approximated with an arbitrarily small number of the same sign as the curvature at the other knot and the test proceeds as before.

In addition to testing two knots for the feasibility of a monotone curvature path, it is sometimes necessary to interpolate such a path. In the figures in this paper, and for measuring the error in using two knots to represent a portion of an image curve, a spline consisting of three circular arcs is used. The spline agrees with the knots at its endpoints in position, tangent direction, and curvature, except when curvature at a knot is zero, in which case its curvature is approximated. The spline is continuous, continuous in tangent direction, and a monotonic step function in curvature: that is, the curvature of the middle arc is between that of the first and last arcs. We shall refer to this spline as the *monotone curvature* spline. See Figure 2 for an example.

V. SMOOTHING WITH GAUSSIANS

In this section the algorithm for finding knots which are candidates for assembly into the final lists is described. The

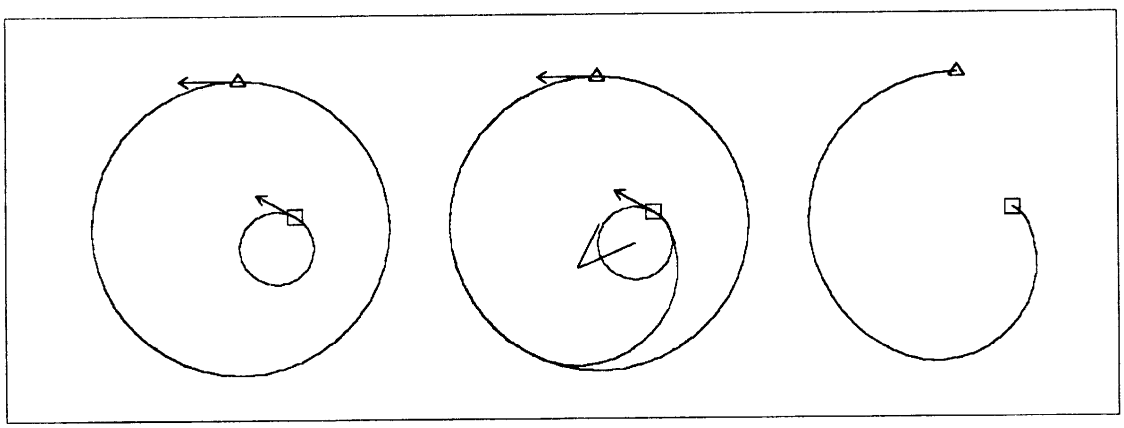

Figure 2: *Monotone curvature splines. Left, two knots which can be interpolated with a monotone curvature path. The square and the triangle indicate the positions, the arrows tangent directions, and the circles curvatures. Center, a monotone curvature spline consisting of three circular arcs interpolates the knots. The first and last arcs coincide with the knots' osculating circles. The vertices of the "V"-shaped polygonal arc are the centers of the three circular arcs. Right, the position markers and the spline are displayed alone.*

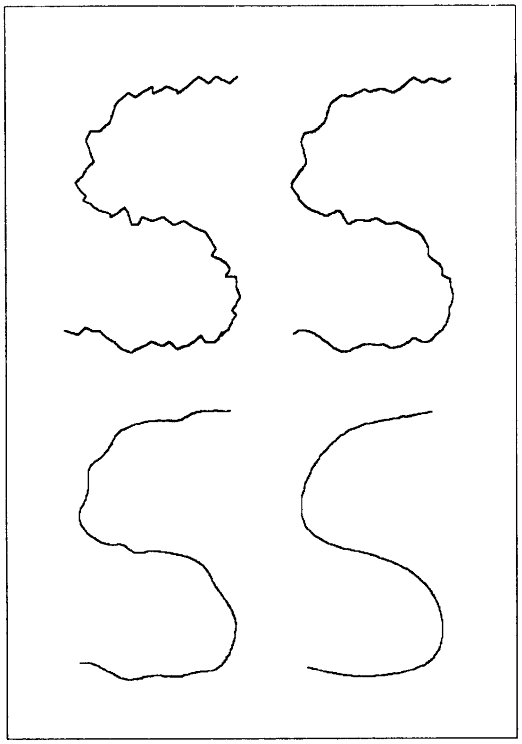

Figure 3: *Smoothing two-dimensional curves with gaussians.* Top left, *a hand-drawn sampled curve. The other curves are smoothed versions of the sampled curve, with the gaussian's scale parameter increasing from* top right, *to* bottom left, *to* bottom right.

goal is to estimate position, tangent direction, and curvature at critical points along the sampled curve. Unfortunately, tangent direction and curvature are not defined for sampled curves. Further, since the goal is to represent the curve at a variety of scales, they must include position, tangent direction, and curvature somehow measured at a variety of scales.

Another constraint is that knots estimated at one scale should be consistent in the sense that it be possible to draw a monotone curvature path interpolating them. This suggests that extracting critical points from curvature estimated by locally fitting circles as in [Brady and Asada 1984] is inadequate for this purpose, since there is no guarantee that the curvature monotonicity relation will hold between adjacent critical points. One way to avoid this problem is to map from the sampled curve to a smooth one and then to detect critical points in the smooth curve.

Smoothing the sampled curve with gaussians at varying resolutions meets these requirements. The smoothing technique discussed in this section produces an infinitely differentiable curve, so that a scale of smoothing defines a map from the sampled curve to a smooth curve (in the sense of infinitely differentiable), and critical points can then be detected in the smooth curve. Varying the scale of smoothing varies the scale at which position, tangent direction, and curvature are measured. Figure 3 is a example of a sampled curve smoothed at several different scales.

[Witkin 1983] has taken this approach in filtering one-dimensional sampled curves. He points out that zero crossings of the second derivative, which are closely related to zeros of curvature, can disappear as the scale of smoothing increases, but new ones can never appear. While we have no corresponding claim for critical points of two-dimensional curves, it is at least intuitively plausible that they exhibit the same behavior, and our experimental evidence is consistent with this conjecture. One desirable consequence is that shorter lists of knots can be used to describe a curve if the scale of smoothing is increased sufficiently.

The basic approach of the smoothing algorithm is to smooth each coordinate function independently after defining it as a function of the straightline distance between adjacent points. At each point, the smoothed value of the coordinate function is a weighted average of the values of the coordinate function at nearby samples; the weights decrease with distance from the point being smoothed. The weighted average is computed by convolving the coordinate function with a gaussian, and normalizing the result at each point to correct for the fact that intersample distances vary along the curve. The normalized result turns out to be infinitely differentiable, so that it is possible to compute position, tangent direction, and curvature of the smoothed curve defined by the two smoothed coordinate functions.

The critical points on the smoothed curve do not necessarily lie at points corresponding to samples of the original curve. The method used to find critical points oversamples the smoothed curve at a rate that depends on the range of intersample distances and computes position, tangent direction, and curvature at each oversampled location. The pattern of sampled curvatures indicate when a critical point lies between samples, and an iterative interpolation method is used to find its location as accurately as necessary. Figure 4 illustrates the critical points of a smoothed curve found by this method.

Given a scale parameter for the gaussian, this algorithm specifies how to obtain a list of critical points, with position, tangent direction, and curvature at each, describing the curve smoothed at that scale. The choice of the range of scales for which smoothing should be performed to obtain these lists has not yet been automated; ultimately it will be based on the range of intersample distances, noise, and expected size of image curve features.

VI. ASSEMBLING KNOTS INTO LISTS

The next step is to assemble the knots obtained from smoothing the curve at different scales into the lists of knots which best approximate the curve. The approximation here refers to some measure of the distance between the original sampled curve and the monotone curvature spline which interpolates the knots on the list. Dynamic programming is used to find for each number of knots the list of knots which best approximates the curve.

Note that scale is used in two senses here. The scale of *smoothing* refers to the scale parameter of the gaussian. The scale of the *representation* refers to the number of knots on a list which approximates the curve. The two may be different because a list of knots output by the dynamic programming algorithm may contain knots obtained from various scales of smoothing.

This is in part a consequence of the definition of approximation error of a list of knots. The error between a consecutive pair of knots and the corresponding portion of the original sampled curve is defined as the area between the monotone curvature spline which interpolates the knots and that portion of the sampled curve. The error for a list of knots is the maximum of these consecutive knot errors. Thus the error for a list bounds the error between any consecutive pair of knots. This reduces the sensitivity of a list to partial occlusion, since the error of most subsets of the list have the same error as the list itself. More global measures of error, like the sum of consecutive knot errors, do not have this property. Thus the representation of subsets of the curve achieving a given approximation error is more likely to be stable with respect to how much of the curve outside the subset is present.

As a portion of the curve is smoothed more and more, the error in using knots obtained from it to approximate the sampled curve on the average increases. But the rate of increase in any region of the curve depends on the behavior of the curve in that region. For example, shallow undulations along a basically linear portion of the curve will result in many knots to capture the small changes in curvature at the smallest scale of smoothing, but perhaps just a knot or two when the scale of smoothing is increasing at a very small cost in increased error in the approximation. At a sharp corner, however, smoothing tends to increase error dramatically as the corner becomes more rounded, but there is no corresponding savings in the number of knots required to describe that portion of the curve.

Thus the tradeoff between error, the number of knots, and their scale of smoothing can vary along a curve. It follows that minimizing the error achieved by a list of n knots can result in knots obtained from different scales of smoothing.

[Plass and Stone 1983] use dynamic programming to find the best list of knots to approximate a sampled curve with parametric cubic splines. The basic idea is to construct a graph which represents all possible lists of knots and to find the minimum error list using the optimal search strategy. Our problem is slightly different, since our goal is to find the best list of knots for each feasible length list. A new algorithm has been developed which finds all such lists in one pass through the graph; Figure 5 displays an example of its output for a curve smoothed at one scale. Each curve is the best approximation to the original curve for its number of knots.

VII. FUTURE RESEARCH

The integration of knots from different scales of smoothing into the same list has in some cases posed problems at those locations on the curve where the optimal scale for the curve is changing rapidly. The likelihood that a monotone curvature transition between adjacent knots will be feasible decreases when the knots are from widely separated scales, since they come from two possibly quite different curves. The current solution is to ensure that the spacing in scales is dense enough to guarantee the possibility of a monotone curvature transition between adjacent knots from different scales. If scale is changing quickly enough even in one part of the curve, this may force smoothing at many scales and therefore generate many sets of candidate knots for the final representation. The dynamic programming technique used to assemble the knots into lists, which performs the (most efficient) *exhaustive* search, has complexity $O(n^2)$ in the number of knots, so clearly an alternative strategy is required. Assembling knots into lists more selectively, by replacing the exhaustive search with one which more cheaply eliminates knots from scales unsuited for representing that portion of the curve from which they were estimated, is one way to make the integration of knots from multiple scales more feasible.

The automation of the selection of natural scales is ongoing. The strategy is to postulate a utility function of the quality and cost of computing with a representation, and choose scales of representation which are local maxima of utility. A preliminary version of this approach has been implemented which uses the approximation error of a list of knots as a proxy for quality, and the length of the list as a proxy for cost. The justification is that approximation error is related to errors in model matching and viewpoint estimation, and the cost of matching and estimation is in part a function of the volume of information on which the required computations are based. So far, the implementation of this approach with simple utility functions has given mixed results, and more work is needed.

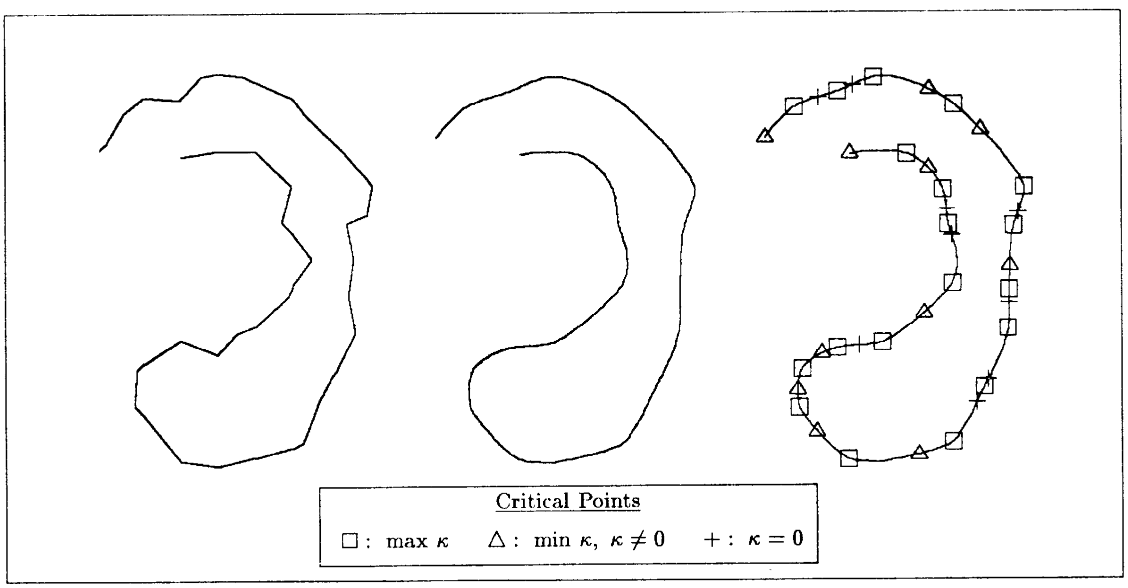

Figure 4: *The critical points of a smoothed curve. Left, a sampled curve produced by a simple edge detection program written by the author and run on a real image. Center, the curve smoothed with a gaussian. Right, the same smoothed curve with critical points marked. Monotone curvature splines interpolate the critical points in the rightmost two figures.*

The ultimate test of the representation will be how well the model-matching and viewpoint estimation algorithm performs using the representation as input. This goal guided the design of the representation, and while the design and implementation of this algorithm is far from complete, it is a crucial part of this research and will be the topic of future papers.

ACKNOWLEDGMENTS

The author thanks Rod Brooks, David Lowe, Brian Wandell, and Andy Witkin for their helpful comments on an earlier draft of this paper.

REFERENCES

[1] Attneave, Fred, "Some informational aspects of visual perception," *Psychological Review,* **61** (1954), 183-193.

[2] Brady, Michael, "Parts description and acquisition using vision," *Proceedings of the Society of Photo-optical and Instrumentation Engineers,* 1982.

[3] Brady, Michael, and Haruo Asada, "Smoothed Local Symmetries and Their Implementation," *The First International Symposium on Robotics Research,* Michael Brady and R.P.Paul, eds., MIT Press, Cambridge, Mass., 1984 (to appear).

[4] Brooks, Rodney A., "Symbolic reasoning among 3-D models and 2-D images," *Artificial Intelligence,* **17** (1981), 285-348.

[5] Duda, Richard O., and Peter E. Hart, *Pattern Classification and Scene Analysis,* Wiley-Interscience, 1973.

[6] Goad, Chris, "Special purpose automatic programming for 3D model-based vision," *Proceedings ARPA Image Understanding Workshop,* 1983.

[7] Hoffman, Donald D., *Representing Shapes for Visual Recognition,* Ph.D. Thesis, Massachusetts Institute of Technology (May 1983).

[8] Hollerbach, J., "Hierarchical shape description of objects by selection and modification of prototypes," *MIT-AI-TR-346,* 1975.

[9] Plass, Michael, and Maureen Stone, "Curve-Fitting with Piecewise Parametric Cubics," *Computer Graphics,* **17**:3 (1983), 229-239.

[10] Witkin, Andrew P., "Scale space filtering," *Proceedings of the Eight Internation Joint Conference on Artificial Intelligence,,* 1983, pp. 1019-1022.

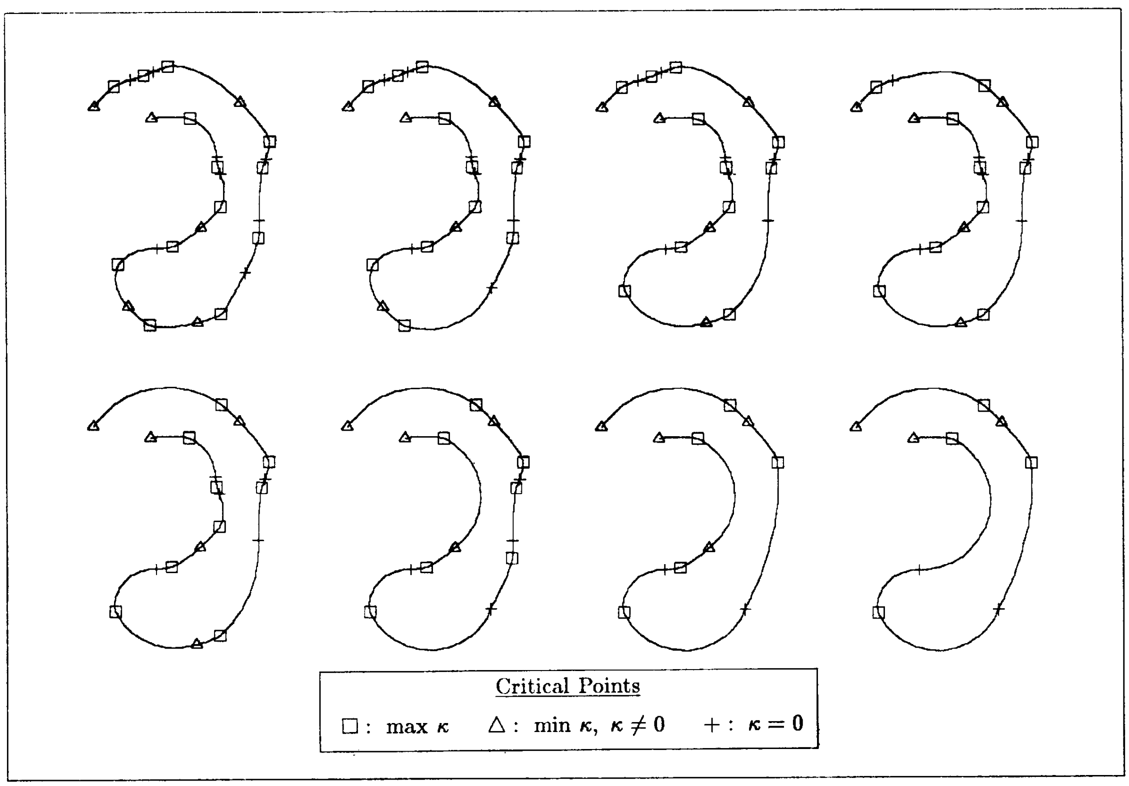

Figure 5: *Finding the best sets of knots to approximate a sampled curve. Each curve above is a set of knots interpolated by the monotone curvature spline. In this example (the same curve as in Figure 4), only one scale of smoothing produced the candidate knots, although the algorithm can handle more scales. A dynamic programming algorithm was used to find the best set of knots to approximate the original sampled curve for each possible number of knots; some of the sets are displayed here. The number of knots decreases most rapidly across rows from left to right and then down columns.*