Proc. 4th Int. Workshop on Visual Form, C. Arcelli, L. P. Cordella, and G. Sanniti di Baja, eds., Lecture Notes in Computer Science, ©Springer-Verlag, 2001

Perception-Based 2D Shape Modeling by Curvature Shaping

Liangyin Yu* and Charles R. Dyer**

Computer Sciences Department
University of Wisconsin
Madison, Wisconsin 53706 USA
{yuly,dyer}@cs.wisc.edu

Abstract. 2D curve representations usually take algebraic forms in ways not related to visual perception. This poses great difficulties in connecting curve representation with object recognition where information computed from raw images must be manipulated in a perceptually meaningful way and compared to the representation. In this paper we show that 2D curves can be represented compactly by imposing shaping constraints in curvature space, which can be readily computed directly from input images. The inverse problem of reconstructing a 2D curve from the shaping constraints is solved by a method using curvature shaping, in which the 2D image space is used in conjunction with its curvature space to generate the curve dynamically. The solution allows curve length to be determined and used subsequently for curve modeling using polynomial basis functions. Polynomial basis functions of high orders are shown to be necessary to incorporate perceptual information commonly available at the biological visual front-end.

1 Introduction

The first goal of visual perception is to make the structure of the contrast variation in the image explicit. For stationary images, the structure is organized through the curvilinear image contours. From the point of view of information theory, the probability that an image contour is formed by some random distribution of contrast is extremely small and thus is highly informative. For the contour itself, it is also more meaningful to identify the part of the image contour that is more informative than other parts of the same contour. Though this principle is important from either the view of information theory or data compression, it is nonetheless essential to inquire about the inverse problem, i.e., how can the less informative part be recovered from the more informative part? This paper is about both problems in the 2D curve domain with main emphasis on the inverse problem.

---

* Current address: Canon Research Centre Europe, 1 Occam Court, Occam Road, Guildford, Surrey GU2 7YJ, United Kingdom

** The support of the National Science Foundation under Grant No. IIS-9988426 is gratefully acknowledged.

Methods for representing 2D curves are usually segment-based with each segment defined by either a straight line (polygon) or a parameterized curve (spline). The segmentation points that separate segments are determined from properties computed along the curve, among them curvature is the most commonly used [4, 5]. However, the properties for curve segmentation are generally highly sensitive to scale changes because the computation is commonly conducted after the contours are identified, which is notoriously dependent on the scale used. This problem can be avoided by using methods that compute the curvature along the contour directly from the image and a carefully designed selection scheme for the scales used in the computation [11].

Curvature has been considered to be one of the major perceptual properties of 2D shapes [2, 5, 10]. It is invariant to rigid transformations and can be computed by our physiological system [3, 7]. It has been used extensively in shape matching [8] and object recognition [6] as well as for shape modeling in both 2D [9] and 3D [1].

From these observations regarding a 2D curve and its perception, the problem of 2D curve modeling can be formulated as a two-stage process: first, the perception-based selection of the local parts on the curve to be modeled, and second, the measurement of relevant modeling parameters regarding the shape. In this paper we also formulate the inverse problem of constructing a curve from a given set of parameters that has been selected previously as modeling parameters. The combination of the significance of curvature in visual perception and the importance of geometrical modeling of image contours is motivation for developing a new framework for 2D curve representation and reconstruction using curvature.

2 Background

2.1 Direct Curvature Computation from an Image

Curvature computation on image contours is generally sensitive to noise due to the way the computation is conducted, i.e., compute curvature from contour detected. This problem can be remedied greatly by computing curvature directly from an image at a tentative contour position [11]. The method can be extended to the computation of higher-order differential invariants such as the derivative of curvature, which will be used extensively in this paper.

Let $I(x, y)$ be the input image. A 2D Gaussian kernel is separable and defined by $\psi_{00}(x, y; \sigma) = \psi_0(x; \sigma)\psi_0(y; \sigma)$, with σ being a scale parameter and $\psi_0(x; \sigma) = (1/\sqrt{2\pi}\sigma)\exp(-x^2/2\sigma^2)$ an 1D Gaussian kernel. The ith-order and jth-order differentiations of ψ with respect to x and y are given by $\psi_{ij}(x, y; \sigma) = \psi_i(x; \sigma)\psi_j(y; \sigma)$. It can be shown that the curvature at location (x_0, y_0) of an image contour is given by

$$\kappa(x_0, y_0) = \frac{\psi_{20}(x^r, y^r) * I(x, y)}{\psi_{01}(x^r, y^r) * I(x, y)} \tag{1}$$

where $*$ is the convolution operator and $(x^r, y^r) = (x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta)$ with θ being the orientation of the contour. The derivative of curvature is then given by $d\kappa/ds = \kappa\lambda - (\psi_{30}(x^r, y^r) * I(x, y))/\Phi_\theta$, where $\Phi(x, y, \theta; \sigma) \stackrel{\triangle}{=} -(\partial\psi_{01}/\partial\theta) * I(x, y)$, $\lambda = -\nabla\Phi \cdot \boldsymbol{n}/\Phi_\theta$, and $\Phi_\theta = \partial\Phi/\partial\theta$ with $\boldsymbol{n}$ being the unit normal vector to the contour. An example of the curvature computed in this way is shown in Figure 1.

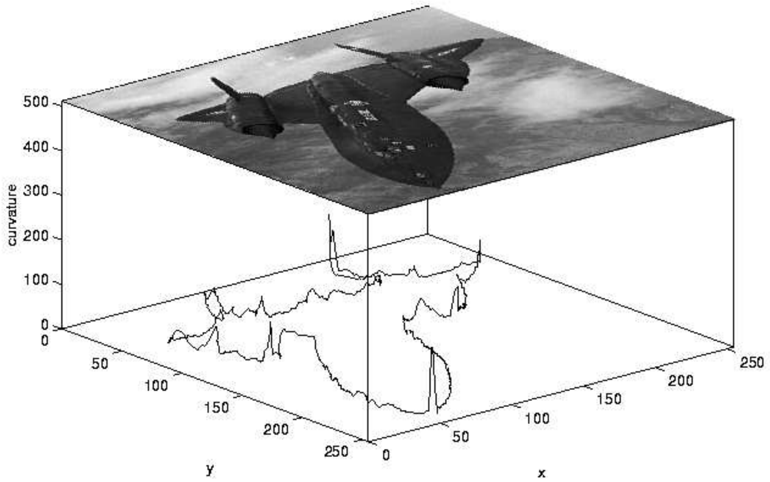

Fig. 1. The curvature along the contour of an airplane image.

2.2 Geometry of 2D Curves

Given a curve $\boldsymbol{c}(s) = (x(s), y(s))$ parameterized by its curve length s, the fundamental theorem of differential geometry for planar curves enables us to describe the curve uniquely (up to a rotation θ_0 and a translation (a, b)) using its curvature $\kappa(s)$. This is explicitly formulated by the intrinsic equations: $x(s) = \int \cos(\theta(s))\, ds + a$, $y(s) = \int \sin(\theta(s))\, ds + b$, $\theta(s) = \int \kappa(s)\, ds + \theta_0$ with three boundary conditions given to specify (a, b) and θ_0. The curve $\kappa(s)$ is the *curvature space* for $\boldsymbol{c}(s)$.

Hence the problem of shape representation in 2D image space is equivalent to the representation of the function $\kappa(s)$ in curvature space. The primary difficulty in using the intrinsic equations directly in curve reconstruction from curvature space is that there is no well-defined computational procedure for constraining the shape in either the 2D image space or the curvature space from the changing shape in the other space. In other words, when taking into account noise, both spaces are unstable by themselves. However, by incorporating both spaces into a reconstruction algorithm, satisfactory results can be achieved.

We will subsequently consider the following geometrical parameters of a curve. Given a smooth curve $\boldsymbol{c}(s)$ that is C^2 continuous, two points P_0, P_1 on $\boldsymbol{c}(s)$ at s_0, s_1 are such that the respective curvatures $\kappa(s_0), \kappa(s_1)$ are curvature extrema, i.e., $\kappa'(s_0) = \kappa'(s_1) = 0$. The points P_0, P_1 are called *feature points* for $\boldsymbol{c}(s)$ (Figure 2).

Given this background, the problem of *2D shape representation using feature points* will be to locate the curvature extrema along a curve and construct the curve using these extremal points. Traditionally this goal is achieved through piecewise interpolation using cubic splines and matching boundary conditions at the knots. However, this approach is unable to incorporate the higher-order constraints of κ and κ'. The other problem is the relatively straight segments provided by this model, requiring more knots for more curved regions. This fact may not be favorable when the scale-space factor is

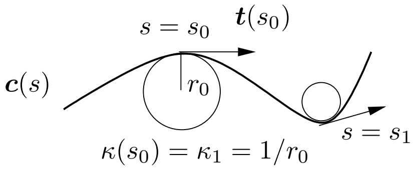

Fig. 2. The geometrical factors that determine a 2D curve.

taken into account, which requires a more or less even distribution of knot positions along the curve at a given scale. These problems can be alleviated by using higher order splines. Another problem is the extra feature points inserted by the basis functions. To solve this problem, a different approach that works on both image space and curvature space is required.

The parameter used for the interpolation is also problematic, especially when curve length parameterization is required. For a given image, the curve length along an image contour can generally be estimated quite accurately, and the relevant geometrical and modeling parameters can be computed. However, the inverse problem of finding the curve from a given set of boundary conditions does not provide information on curve length. The method presented in the next section provides the curve length information as one of its results. This information can then be used for modeling the curve using the high-order polynomial basis functions presented in Section 4.

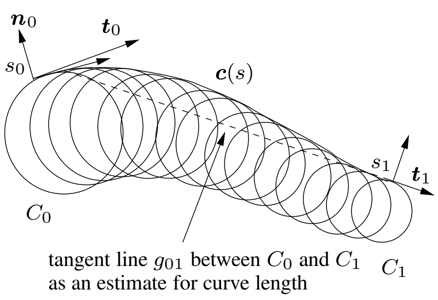

Fig. 3. Dynamical moves for a curve segment between two convex points.

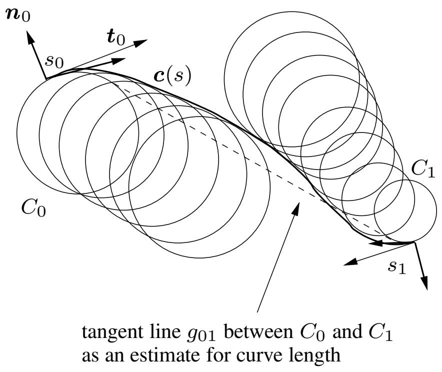

Fig. 4. Dynamical moves for a curve segment between one convex and one concave point.

3 Curve Representation by Curvature Space Shaping

Given two feature points, P_0, P_1, on an unknown curve $\boldsymbol{c}(s)$ in an image and their associated tangent orientations, θ_0, θ_1, as well as their signed curvatures, κ_0, κ_1, we now present a method to solve the problem of finding the curve that satisfies the given boundary conditions with the property that there is no computable curvature extremum in between P_0 and P_1 other than those at P_0 and P_1.

Let the osculating circle at P_0, P_1 be C_0, C_1 respectively, and the tangent line between C_0, C_1 in the direction of $\boldsymbol{t}_0 = (\cos\theta_0, \sin\theta_0)$ be g_{01} (the unit vector $\boldsymbol{g}$ along g_{01} at the C_0 end has the property $\boldsymbol{g} \cdot \boldsymbol{t}_0 > 0$). Among the four tangent lines, g_{01} is chosen to be one of the two non-crossing ones closest to P_0 if $\kappa_0\kappa_1 > 0$, and one of the two crossing ones if $\kappa_0\kappa_1 < 0$. The curve $\boldsymbol{c}(s)$ is constructed by dynamically moving stepwise from P_0 along a direction that will gradually changed into the direction of $\boldsymbol{t}_1 = (\cos\theta_1, \sin\theta_1)$ while gradually changing the curvature of corresponding osculating circle in the process (Figures 3 and 4). Since $\boldsymbol{c}(s)$ is unknown, the curve length between P_0 and P_1 cannot be determined in advance. Rather, we use the length of the tangent line g_{01} as an initial estimate for the curve length.

Let the desired steps for reaching P_1 from P_0 be n, and let the length of tangent line g_i at each step be s_i^g for $i = 1, \ldots n$. The direction of movement at each step i is determined by the corresponding θ_i and $\boldsymbol{g}_i$ by $(\boldsymbol{t}_i + \boldsymbol{g}_i)/2$, i.e., move half way in between $\boldsymbol{t}_i$ and $\boldsymbol{g}_i$. The curvature κ_i of the osculating circle C_i is given by $\kappa_0 + (\kappa_1 - \kappa_0)/n$. The distance d_i to be moved consists of a movement along $\boldsymbol{g}_i$ followed by a movement along C_i and is given by $d_i = s_i^g/(n-i+1)$. The orientation change caused by this movement is then given by $\varDelta\theta = \kappa_i d_i = (\kappa_1 - \kappa_0)d_i/n$. This is formulated in such a way that if the estimated curve length s_i^g is indeed the curve length, then at each step we will move precisely $1/(n-i+1)$ of the curve length and will complete our journey in n steps. The curve length is thus given by

$$s = \sum_{i=1}^{n} \frac{s_i^g}{n-i+1}$$

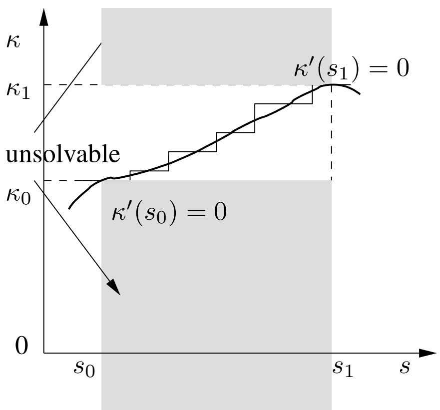

Fig. 5. The curvature space for a curve segment with two convex points.

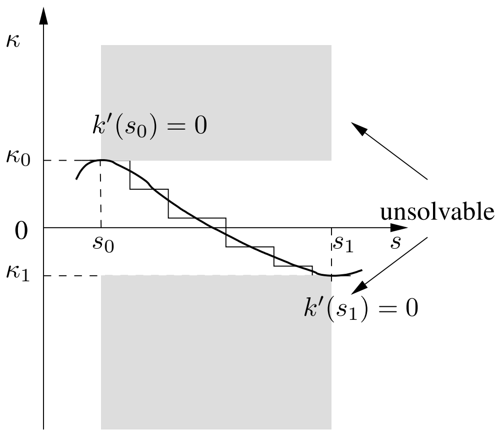

Fig. 6. The curvature space for a curve segment with one convex and one concave point.

The corresponding curvature spaces of the curves in Figures 3 and 4 are given in Figures 5 and 6, respectively. Under the solvability conditions explained in the next section, it can be shown that the movement will approach P_1 in n steps with the desired boundary conditions, and the following limits can be established:

$$\lim_{n\to\infty} P_n = P_1, \qquad \lim_{n\to\infty} \theta_n = \theta_1, \qquad \lim_{n\to\infty} k_n = k_1$$

Each of the n segments of curve $\boldsymbol{c}(s)$, according to the curvature space, is a partial arc on the osculating circle C_i with constant curvature κ_i (Figure 7), which can be approximated by a piecewise straight segment. Even though the curves have great similarity, their curvature spaces have completely different shapes. This illustrates the difficulty in working from only one of the spaces. In comparison, a constant curvature segment can better track the tangent line and converge faster to the destination than the straight segment counterpart because the osculating circle at each point bends toward rather than away from the line. This implies fewer steps are required and better precision.

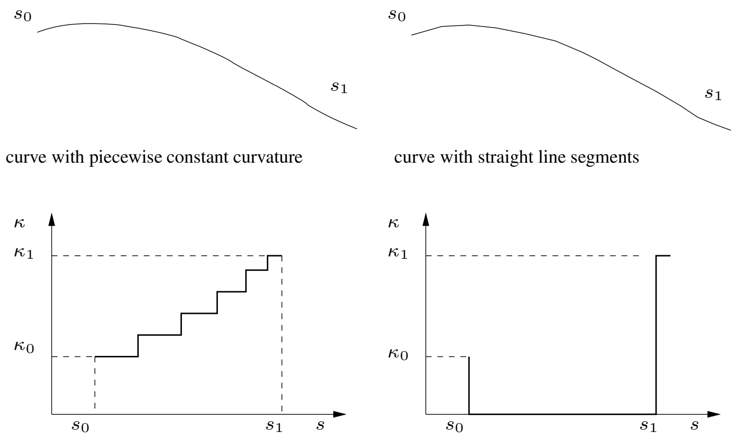

Fig. 7. The two segment models of a curve.

3.1 Solvability Conditions

There are two conditions governing whether this problem has a solution. One corresponds to the "sidedness" of the object, since the sign of curvature is defined according to which side of the object the normal vector lies by the Frenet equation $\boldsymbol{t}' = \kappa \boldsymbol{n}$. The tangent line g_i gives an estimate of the curve length of $\boldsymbol{c}(s)$ and at the same time defines on which side the object lies. It is unsolvable when the boundary conditions create an impossible object. The other condition of solvability is whether during the process the curvature enters an area in which extra extrema have to be created. These areas are denoted *unsolvable* in Figures 5 and 6. These two unsolvability conditions are illustrated in Figure 8.

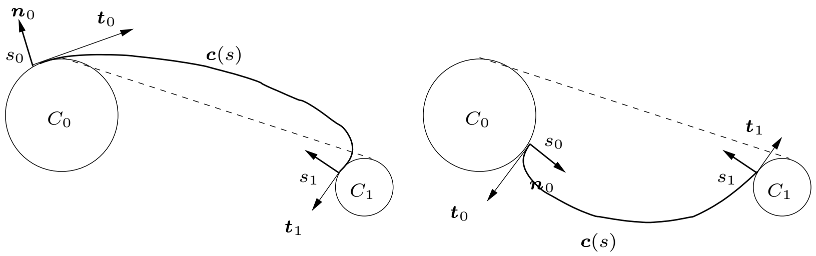

Fig. 8. The two unsolvability conditions between two convex points.

4 Curve Representation by Polynomial Basis

The curve length of an arbitrarily parameterized curve $\boldsymbol{c}(t)$ is $s = \int |\boldsymbol{c}'(t)|\, dt$. From this formulation it is clear that if $\boldsymbol{c}(t)$ is represented by polynomial basis functions, its curve length will not be polynomial, and vice versa. Hence, to use a polynomial basis we can either work in the image space of $\boldsymbol{c}(t) = (x(t), y(t))$ by fitting the boundary conditions $\boldsymbol{c}_0, \boldsymbol{c}_1, \theta_0, \theta_1, \kappa_0, \kappa_1, \kappa_0', \kappa_1'$, or work in curvature space through the intrinsic equations on the same set of boundary conditions. Both methods result in a set of highly nonlinear equations with the existence of a solution questionable. In this section we introduce a compromise method using polynomial basis functions in image space that satisfy all the boundary conditions but do not guarantee that new curvature extrema will not be inserted in the models.

4.1 Curves from Hermite Splines

The Hermite polynomials $H_{i,j}(t)$ of order L with $i = 0, \ldots, L$ and $j = 0, 1$ satisfy the *cardinal property*: $H^k_{i,0}(0) = \delta_{ki}, H^k_{i,0}(1) = 0, H^k_{i,1}(0) = 0, H^k_{i,0}(1) = \delta_{ki}$, where the superscript k indicates the order of differentiation. This set of equations defines polynomials of order 3 for $L = 1$, of order 5 for $L = 2$, and of order 7 for $L = 3$.

The cardinal property is useful for fitting boundary conditions of various differential orders. We will consider here the case of first, second and third order differentials. Let P^i_j be the ith derivative of the curve at $\boldsymbol{P}_j = \boldsymbol{c}(t_j)$. The curve segment connecting the two points ($t = t_0, t = t_1$) using Hermite splines is

$$\boldsymbol{c}(t) = \sum_{j=0}^{1} \sum_{i=0}^{l} \boldsymbol{P}^i_j H_{i,j}\left(\frac{t - t_0}{t_1 - t_0}\right), \qquad l = 1, \ldots, 3$$

This formulation is given in terms of the differentials at two boundary points, which are not readily available since the problem is given the conditions of location ($\boldsymbol{P}_0, \boldsymbol{P}_1$), orientation ($\theta_0, \theta_1$), curvature ($\kappa_0, \kappa_1$), and differential of curvature (κ_0', κ_1'). Since the estimation of $\boldsymbol{P}^i_j$ is generally noisy, it is necessary to use curve length as a parameter. Hence, the curvature $\kappa = x'y'' - x''y'$ and $\kappa' = x'y''' - x'''y'$ since $(x')^2 + (y')^2 = 1$. Given κ and κ' allows us freedom to choose two additional conditions to fix x'', y'' and x''', y'''. This can be done arbitrary since the problem itself does not dictate these conditions.

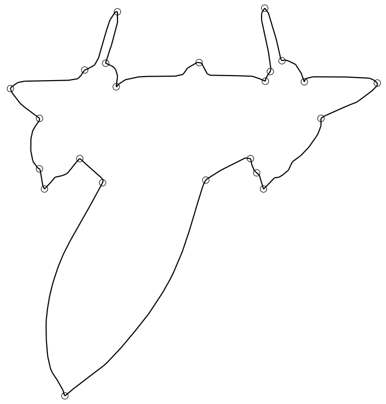

Fig. 9. The image contour of the airplane in Figure 1.

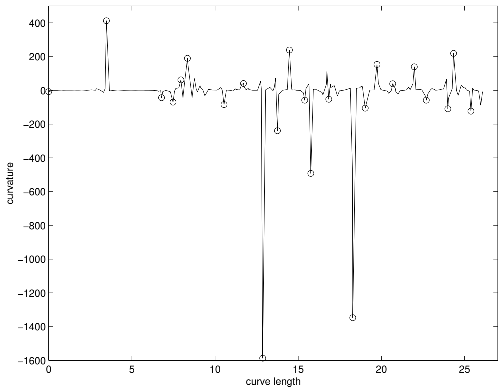

Fig. 10. The curvature space and extrema for the airplane contour.

5 Examples

For the airplane contour in Figure 9, the curvature space is given in Figure 10, in which the curvature extrema are identified. The corresponding feature points are also marked in Figure 9. These points are the component partition points in which the concave points separate components while the convex points mark the partition of different segments within the same component. One of these components with feature points within the segment identified is shown in Figure 11. The curve length is estimated from the curve computed using the method of curvature shaping, and subsequently used in the representation by Hermite basis functions. Three different orders of the basis functions were used. Bases of order 3 used the location and tangent orientation information only. Bases of order 5 also matched boundary conditions for the curvature, while Hermite bases of order 7 satisfied the additional condition that these points are actually feature points with extremal curvature.

6 Discussion

6.1 Scale Space

Scale space manifests its effect mostly in computation. From the formulation in Section 2.1, it can be observed that after the image contours are computed, the effect of the 2D scale-space kernel parameterized by rectangular Cartesian coordinates is equivalent to the effect of an 1D scale-space kernel parameterized by curve length. This is because the image contour is computed by orienting the kernel ψ_{01} in the direction of the contour [11]. This essentially creates a "curvature scale space" [9], in which variations within a fixed scale are gradually lost when the scale getting coarser. This results in a separation of feature points on the curve with a distance proportional to the scale used for the computation. Hence, even though the curvature $\kappa(s)$ is a highly nonlinear

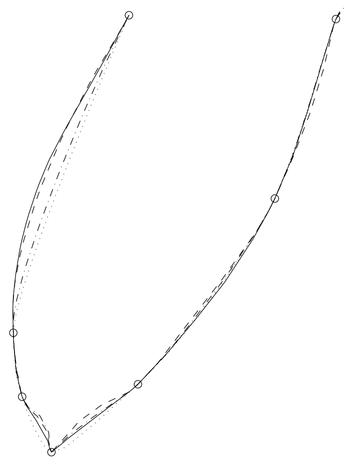

Fig. 11. Parts of the airplane represented by Hermite basis of order 3($\cdots$), 5 ($-\cdot$),7 ($--$), compared to the original (—).

function of $x(s)$ and $y(s)$ and its shape cannot be exactly predicted for a given scale, the feature points with extremal curvature can nonetheless be located by searching the curvature space of finest scale and partitioning the curve with segments of length proportional to the scale without actually computing the curvature scale space for coarser scales.

6.2 Perceptual Boundary Conditions

The measure of distance in biological visual perception is provided by comparison between a reference length and what is to be measured, i.e., there is no intrinsic metric. This renders computations using distance measure (such as optical flow and curvature) imprecise. On the other hand, an orientation measure has built-in mechanisms with a certain degree of precision. From these observations, the primary measurement of the local shape of a curve will be the position and orientation relative to a 2D Cartesian coordinate system, while curve length and curvature are much less precise in terms of measurement. Hence the primary boundary conditions related to perception are locations and orientations. However, we do show that when secondary boundary conditions such as curvature and its derivative are available, the representation is much more compact and precise. For example, by using cubic or quintic Hermite bases for the component in Figure 11, precision can be augmented by adding more knot points. However, it is not clear which ones to choose since there is no special attribute in curvature space to facilitate this choice.

6.3 Component Partitions Using Curvature Space

There are two different kinds of information presented through the curvature space when the whole contour is considered. The prominent ones are actually component partition points (negative extrema) of the shape or segment partition points (positive extrema) within a component. This can be seen in Figures 1 and 10. Feature points for each segment have to be identified within the segment rather than compared to every segment in the contour, and their extremal property is essential to the modeling. The inadequacy of lower-order Hermite bases to represent a curve segment is clearly seen in Figure 11, since these do not take into account the extremal property at these points.

7 Conclusions

A compact description of a smooth curve was presented in this paper, based on curve features that are perceptually-important. The geometrical model of these features was defined by location, orientation and curvature, with the additional property that the curvature reaches extremal values at feature points. Being able to describe compactly an object contour is of great importance in object recognition, especially when the description is independent of the viewpoint. We also develop a method to identify the curve from a given set of perception-based boundary conditions at prescribed feature points. One of the results is a good estimation of curve length that can subsequently be used by polynomial basis functions for curve modeling. However, to satisfy the given boundary conditions, much higher-order polynomials are needed than what are commonly used.

References

[1] H. Asada and M. Brady. The curvature primal sketch. *IEEE Trans. Patt. Anal. Machine Intell.*, 8(1):2–14, 1986.

[2] F. Attneave. Some informational aspects of visual perception. *Psychology Review*, 61:183–193, 1954.

[3] A. Dobbins, S. Zucker, and M. Cynader. Endstopping and curvature. *Vision Research*, 29(10):1371–1387, 1989.

[4] M. Fischler and R. Bolles. Perceptual organization and curve partitioning. *IEEE Trans. Patt. Anal. Machine Intell.*, 8(1):100–105, 1986.

[5] M. Fischler and H. Wolf. Locating perceptually salient points on planar curves. *IEEE Trans. Patt. Anal. Machine Intell.*, 16(2):113–129, 1994.

[6] D. Hoffman and W. Richards. Parts of recognition. *Cognition*, 18:65–96, 1985.

[7] J. J. Koenderink and A. J. van Doorn. Receptive field families. *Biological Cybernetics*, 63:291–297, 1990.

[8] E. Milios. Shape matching using curvature processes. *Computer Vision, Graphics, and Image Processing*, 47:203–226, 1989.

[9] F. Mokhtarian and A. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Patt. Anal. Machine Intell.*, 14(8):789–805, 1992.

[10] W. Richards, B. Dawson, and D. Whittington. Encoding contour shape by curvature extrema. *J. Opt. Soc. Amer. A*, 2:1483–1491, 1986.

[11] L. Yu and C. Dyer. Direct computation of differential invariants of image contours from shading. In *Proc. Int. Conf. Image Processing*, 1998.