COMPUTER VISION AND IMAGE UNDERSTANDING
Vol. 68, No. 2, November, pp. 170–189, 1997
ARTICLE NO. IV970533

Shape Representation and Recognition from Multiscale Curvature

Gregory Dudek*

Centre for Intelligent Machines, McGill University, Montreal, Quebec, Canada H3A 2A7

and

John K. Tsotsos†

Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 1A4

Received March 31, 1994; accepted July 15, 1996

We present a technique for shape representation and the recognition of objects based on multiscale curvature information. It provides a single framework for both the decomposition and recognition of both planar curves as well as surfaces in three-dimensional space. The decomposition operation simultaneously performs data interpolation, data smoothing, and segmentation. The unification of these three stages results in a smoothing operation that is coupled with the primitives to be used in description. Each of the minimization operators, in addition to having a curvature tuning, also has a different spatial sensitivity function. As a result, the different possible descriptions capture information at multiple spatial scales. This allows a single region of an object to be described in more than one way, when appropriate. The practicality of the ensuing representation is demonstrated by the recognition of planar curves. A matching strategy based on dynamic programming is used. The results illustrate the manner in which a continuous spectrum of similar objects can be defined, ranging from those that are very similar to a target to those that are very different from it.

1. INTRODUCTION

In this paper, an approach is proposed for describing objects for the purposes of recognition. We deal with two issues: building general-purpose multiscale descriptions of *curved objects,* and extracting a concise set of attributes that can be used for recognition. Typical examples of the types of curve we are able to describe as both qualitatively similar, yet discriminably different, are shown in Fig. 1. The method is based on the use of approximating curves with simple curvature properties. In contrast, alternative methods based on line segments or curvature extrema alone are likely to find three of these four shapes indistinguishable, while methods such as shape templates may be oblivious to their similarity.

Much evidence from both psychology and computer vision suggests that the estimation of curvature appears to be important for several tasks such as object modelling and recognition [3, 43]. The stable extraction and measurement of curvature information in the presence of noise has been dealt with in several ways [6, 12, 31, 62, 65, 67]. One characteristic of most existing curvature-measurement techniques is the assumption that there is a unique curvature measurable at each point on an input curve.[1] While this is obviously true in the analytic case, this leads directly to the issues of signal-versus-noise and of scale in the context of computer vision. Despite the respectable results achieved by some researchers, the need for scale-specific operators to deal with noise problems causes an inherent preference for certain ranges of curvature value and involves strong implicit assumptions about the underlying signal. The curvature of a signal depends on what we call noise and what we call meaningful signal, and hence may take on differing values depending on our goals.

We propose that smoothing processes should explicitly take into account the particular measurements to be made from the smoothed data, specifically, the multi-scale measurement of curvature. The **curvature-tuned smoothing** method we have developed and present here allows us to obtain measurements which have not been subjected to an unnatural ''flattening'' or distortion as a result of the smoothing [13]. Our technique is based on multiscale smoothing with a specialized form of ''regularization.'' We

* Corresponding author. E-mail: dudek@cim.mcgill.edu.
† E-mail: tsotsos@vis.toronto.edu.

[1] This assumption is often only stated implicitly. Some techniques use multiple scales at an intermediate filtering stage, but then concentrate on selecting the single correct curvature at each point.

1077-3142/97 $25.00

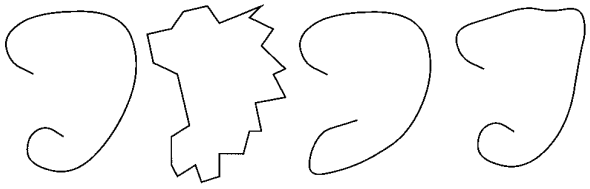

FIG. 1. Discriminable, yet related, curves. All these curves are similar, yet different enough to be discriminable. Many existing curve matching techniques will have difficulty with them, due to their amorphous structure.

show how that can also be followed by a feature extraction stage that provides a representation suitable for recognition of curved objects. The fundamental goals and attributes of this approach are the following:

- To use curvature as a fundamental descriptive attribute.
- To provide a representation that can deal with regions for which the choice of ''true'' curvature is scale dependent.
- To target the smoothing technique to the primitive or measurable to be used in recognition.
- To handle noise in a natural and automatic manner.
- To obtain abstract (i.e., non-pointwise) descriptions of objects.

The method is based on computing alternative smooth descriptions of an input curve using a parameterized smoothing functional. The process inserts discontinuities while performing smoothing; the sections between discontinuities are rated in terms of their goodness-of-fit with respect to the smoothing term and described with a small number of parameters. The use of these parameters for indexing is then illustrated.

1.1. *Background*

Several methods have been developed to segment curves and extract stable features, sometimes based on curvature. Typical approaches include measuring curvature properties, perhaps attempting to find the single most appropriate measurement scale at any point on the curve. These initial measurements are then used to label points or segments of the curve. Such labelled features can then be used for curve measurement if a similarity metric is provided. Several alternative methods within this general paradigm have been developed [1, 2, 4, 23, 28, 34, 36, 44, 49, 63, 66].

Coupled to the question of how to measure curvature is that of how to organize primitive measurements or how to abstract raw curvature data. Asada and Brady have developed a description they refer to as the ''curvature primal sketch'' [2]. This is an extension of Marr's concept of the ''primal sketch'' as a fundamental and comprehensive intermediate image representation. The description is a multiscale structure based on the extraction of changes in curvature. From the curvature features, a description of the contour in terms of structural primitives is constructed: ends, cranks, etc. Ettinger further extends this to deal with more complex domain-specific primitives [15]. One of the unresolved questions with this approach is that of the stability of the structures, and the question of their expressive sufficiency and ubiquity.

Codons are a descriptive primitive based on curvature features motivated, in part, by human perception [21, 47]. Codons are defined by sequences of curve extrema, separated by curvature minima. By enumerating the set of combinations of such extrema, a simple ''vocabulary'' for curves is developed. Relational constraints that further limit the possible combinations of codons constrain the vocabulary and increase the redundancy in the encoding; this can be perceived as advantageous because it permits error correction (as pointed out by the authors); on the other hand, it reduces the efficiency. Although the codon description introduces a conceptually appealing part decomposition of curves and can be used for certain types of recognition task [48], several important computational issues including its robust extension to multiple scales and three dimensional (surface) data remain to be fully explored.

Saund constructs a scale space of edge elements by avoiding Gaussian blurring to construct a scale space hierarchy directly from simple edge fragments [51, 52]. The structuring principle is that of aggregating successively larger or more complex tokens using a predefined set of grouping strategies. This structuring paradigm is in contrast to numerical filtering methods and is, in some ways, an analogue of discrete knowledge organizing techniques for visual information such as ''part-of'' hierarchies [59, for example]. A somewhat similar approach is pursued by Lowe and by Binford [30, 32, 33]. This approach is based on the aggregation of primitive elements (points, edge fragments) using ''perceptual grouping'' principles that exploit quasi-invariant geometric principles.

It has been shown that the extraction of curvature inflection points from a Gaussian scale-space can be used to recognize curved structures [38, 39, 60, 61]. One difficulty with this approach, however, is that curves or subcurves without inflection points all fall into the same equivalence class. Kimia [25, 26] has developed an interesting method for curve decomposition based on erosion-diffusion space. It is based on a combination of smoothing and erosion to extract subpart decompositions of objects which may prove to be useful in recognition.

To summarize the status of curve description methods (and much work on surface recognition), various approaches have been developed that can be effective given a suitable (or readily apparent) decomposition of the curve

into subparts. Likewise, segmentation schemes exist but many are either difficult to relate to *stable* feature extraction or multiscale processing. Curvature has proven to be a singularly important descriptive feature attribute for both computational and psychophysical reasons. A key issue is how to deal with the ''scale problem'' and how to extract usable descriptive tokens within a framework that combines the desirability of curvature-based cues, scale, segmentation and matching.

1.2. *Overview*

Curvature-tuned smoothing is a representation that can be computed for both curves and surfaces and allows regions with different curvatures to be extracted as alternative descriptors at a single location on a curve or surface. The multiscale nature of the representation provides a description that can be made robust to noise because it is based on smoothing yet is also sensitive to structure at multiple scales. In this work we presuppose the availability of a set of points defining a plane curve $\mathbf{d}(t)$, or points in 3-space defining a surface $d(x, y)$.

The equivalence classes of curves generated by this method can be made smaller and more discriminative (i.e., the representation in **"curvature space"**) than is the case for representations based on solely the sign of curvature (or signs of mean and Gaussian curvature on surfaces). For example, curves, subcurves, or surfaces without curvature sign changes can be discriminated so long as the curvature varies between them (which must be the case since curvature is a complete description of shape information). This allows for more precise shape discrimination than is uaually associated with curvature based object recognition, particularly in the case of three-dimensional surfaces.

We will begin by reviewing basic concepts of curve representation. Next, in Section 3, we present the curvature-tuned smoothing method for curves and its extension to surfaces in three dimensions. In Sections 4 and 5 we consider the analytical properties of the process and its convergence. A critical next stage is to simplify the multiscale description; we propose this abstraction in Section 6. Finally, we show how this description can be applied to curve matching, present some experimental results and suggest some additional issues.

2. WHY CURVATURE?

Curvature ($\kappa(t)$ for a curve) has desirable computational and perceptual characteristics [1, 3, 27, 29, 42, 50, 65]. One such property of curvature is that it is invariant under planar rotation and translation of the curve [10]. Thus, if $\mathcal{M}: \Re^3 \rightarrow \Re^3$ is a rigid transformation and $\mathbf{u}(t)$ is a parameterized curve, then

$$\int_b^a |\kappa(\mathbf{u}(t))|\, dt = \int_b^a |\kappa(\mathcal{M} \circ \mathbf{u}(t))|\, dt. \tag{1}$$

Curvature is computed from dot and cross products of parametric derivatives and these are purely local quantities, hence independent of rotations and translations. The dot and cross products are based only on the lengths of, and angles between vectors and, hence, are also independent of rigid transformations of the coordinate system.

For a curve in the plane

$$\mathbf{u}(t) = (x(t), y(t)) \tag{2}$$

the curvature can be expressed as

$$\kappa_u(t) = \frac{x'(t)y''(t) - y'(t)x''(t)}{\sqrt{x'(t)^2 + y'(t)^2}^3}. \tag{3}$$

Since curvature and torsion together uniquely define a curve, the knowledge of the curvature function alone is sufficient, in principle, to determine a planar curve.

Despite this simple definition, the measurement of curvature values that are *useful for recognition* is a difficult problem. In principle, curvature is a purely local attribute. In order to cope with noise, however, traditional approaches to the measurement of curvature for plane curves involve the use of nonlocal models (i.e., to stabilize the inverse problem). The extent and shape of the neighborhood used for such processing asserts an implicit scale specificity (as a result of the interpolant or the support function used in estimation).

3. CURVATURE, AMBIGUITY, AND SMOOTHING

One effective method for modelling visual data is by analogy to the energetics of elastic surfaces [58]. Such an energy minimization approach can be used to solve the problems of smoothing, data interpolation, and discontinuity detection. A simplified expression for the bending (potential) energy of an elastic rod of length L is [54]: $E = \int_0^L \kappa_u(t)^2\, dt$ (where we consider, for simplicity, the one-dimensional case). The constraint due to the data, modelled as ideal springs distributed under a bending rod, is given by $E = \int_0^L (u(t) - d(t))^2\, dt$, where $d(t)$ is the rest position of the ends of the springs and $u(t)$ is the location of the curve modelled as a flexible rod. For cases where the deflection is small (i.e., the data is flat), these two terms can be combined and approximated as

$$E(u) = \int_T (u(t) - d(t))^2 + \lambda(u''(t))^2\, dt. \tag{4}$$

This is known as the ''*thin-rod*'' regularizing equation [8, 22, 56, 58, 64], where the second term is referred to as the

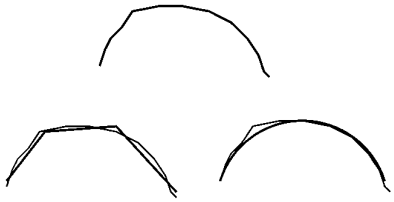

FIG. 2. Ambiguity of "natural" descriptions. Is the data on top "really" a set of straight lines, as shown on the left by the thick lines, or a circular arc, as shown on the right?

stabilizer, in general $\lambda S(u(t), t)$. The equivalent *thin-plate* regularizer for a surface S is given by

$$E(u) = \int_S (u(x, y) - d(x, y))^2 + \lambda(\nabla^2 u(x, y))^2 \, dx \, dy. \quad (5)$$

To this we add appropriate boundary conditions.

In finding a piecewise smooth description of sensed data describing a curve embedded in two dimensions, the discontinuities that are inferred and the curvature estimates produced depend on the particular smoothing model used. One way to perform smoothing is by minimizing a functional of the form

$$E(u) = \int_{t_a}^{t_b} |\mathbf{u}(t) - \mathbf{d}(t)|^2 + \lambda S(\mathbf{u}(t), t) \, dt. \quad (6)$$

By virtue of the fact that it consists of a single set of possible depth values and discontinuities (and hence curvatures) the function $\mathbf{u}(t)$ which minimizes such an energy functional constitutes a commitment to a single *interpretation* of the original data, $\mathbf{d}(t)$. The functional

$$E(u) = \int_{t_a}^{t_t} |\mathbf{u}(t) - \mathbf{d}(t)|^2 + \lambda \kappa_u(t)^2 \, dt \quad (7)$$

selects the lowest curvature interpretation of the original data while rejecting alternative (higher mean squared curvature) interpretations even though they are equally plausible based on either physical or perceptual constraints (for example, see Fig. 2).

Minimum curvature approximations of noisy data, however, are not necessarily the most appropriate. We suggest that the customary preference for surfaces of near zero curvature, despite its mathematical simplicity, may be unnecessarily conservative as a model of either the natural world or human perception. In fact, when the surface data is intrinsically curved, the use of the conventional thin beam smoothing process (or any functional with a linear Green's function) may make it difficult to smooth noisy data sufficiently to extract measurements such as curvature without seriously corrupting them.

For example, the thin-plate smoothing kernel for a bi-infinite plane has a smoothing kernel given by [8]

$$G(t, t_0) = \frac{1}{2\lambda^{1/4}} e^{-|t-t_0|/\lambda^{1/4}\sqrt{2}} \cos\left(\frac{|t - t_0|}{\lambda^{1/4}\sqrt{2}} - \pi/4\right) \quad (8)$$

which over large distances can be approximated by its envelope, a symmetric exponential decay, $G(\alpha, t) = e^{-\alpha|t_0 - t|}$, with $\alpha = 1/\lambda^{1/4}\sqrt{2}$. This is a low-pass filter whose attenuation can be expressed as the function $R(\omega) = 2\alpha/(\alpha^2 + \omega^2)$ in the frequency domain, that is, asymptotically proportional to spatial frequency. For sinusoidal input data in one dimension given by $d(t) = A \sin((1/A)t)$, the precise curvature is

$$\kappa_d(t) = \frac{(1/A)\sin(1/A)t}{\sqrt{1 + \cos^2(1/A)t}}. \quad (9)$$

If we smooth this with a kernel of the above form to reduce nose, for example with a kernel $ge^{-\alpha t}$ (where g is a constant), the measured curvature is distorted to be

$$\kappa(G(\alpha) * d(t)) = \left\{\frac{-g\alpha^2}{\alpha^2 + 1/A^2}\right\} \kappa_d(t). \quad (10)$$

This difficulty applies similarly to any realizable linear filter since a sinusoid is the eigenfunction for linear filtering. Hence, the effect of regularization or filtering will appear as a perturbation to the curvature function whose amplitude is proportional to the amplitude of the transfer function at the sinusoid's frequency. Oliensis has considered the use of a smoothing filter that can perform smoothing with limited distortion, but it operates at a single scale and requires foreknowledge of the appropriate smoothing scale to synthesise a suitable matched filter [41].

Since other contours with moderate curvature also appear to be natural descriptions, as illustrated in the previous figure, why should functions be smoothed towards a curvature of zero? For curves forming the boundaries of curved objects, zero curvature may be a less likely *a priori* estimate since the entire object must have a convex contour at a sufficiently coarse scale, and hence a mean positive curvature.[2] It can be argued that for a closed curve (an object) with area a the most neutral assumption is that it is a circular blob. In this case, a reasonable default assumption for the mean curvature of such an object is[3] $\sqrt{\pi/a}$. In

[2] We will arbitrarily define positive curvature to be convex, with full generality. The sign of the curvature function is arbitrary and its assignment is immaterial.

[3] Since the most neutral assumption for the shape of a closed object is a circular blob, if the area is $a = \pi r^2$, then the radius is roughly $\sqrt{a/\pi}$.

general, of course, an object's contour is not a single uniform circle and this default assumption is far from consistent with reality. What is a natural *a priori* curvature to assume in this case?

Regardless of what *a priori* choice we make for natural curvatures of objects, it will be inappropriate for a wide range of structures that are not consistent with it. It is possible, however, to construct a *continuum* of smoothing functionals

$$E(u, c) = \int_{t_a}^{t_b} \|\mathbf{u}(t) - \mathbf{d}(t)\|^2 + \lambda S(\mathbf{u}(t), c)^2 \, dt \qquad (11)$$

that allow us to extract *multiple* interpretations of the original data as a function of c. The minimization of this functional can be performed for various values of c producing different results $\mathbf{u}(t)$. If we choose $S(\mathbf{u}(t), c)$ to have the form $(\kappa_u(t) - c)$ then c *tunes* the curvature constraint and the minima of the family of functionals given by

$$E(u, c) = \int_{t_a}^{t_b} \|\mathbf{u}(t) - \mathbf{d}(t)\|^2 + \lambda(\kappa_u(t) - c)^2 \, dt \qquad (12)$$

specifies the *set* of functions $\mathbf{u}(t)$ that correspond to uniform curvature approximations of the original data. The application of this set of functionals, collectively, is what we define as **curvature-tuned smoothing** (CTS) and each functional alone is a **layer** of the process. For any functional $E(u, c)$, the solution function $\mathbf{u}(t)$ is the smooth approximation of the original data under a different *a priori* assumption, c, of what the curvature should be. As such, the constant c will be referred to as the **target curvature.** In principle, this tuning parameter is a continuous variable that varies from $-\infty$ (infinitely concave) through zero (flat) to $+\infty$ (infinitely convex). In practice, we will use this parameter to discretely sample curvature space at fixed values denoted by c_i.

If the target curvature is equal to zero ($c_i = 0$) the effects are very much like those of the approximation for the thin-plate smoothing model (using the "quadratic variation" [19]). Our solution (Eq. (12)) expresses a more general approximation to the energy of a rod under tension than the conventional linearized model, but behaves similarly. When the solution approaches being perfectly flat, these two solutions converge. In contrast, the case for $c_i \gg 0$ corresponds to an attempt to fit a *convex* solution to the original data, while the solution for $c_i \ll 0$ corresponds to a *concave* fit.

The differing possible values for the target curvature constitute *models* for the fitting process. In essence, the choice of any specific target curvature corresponds to a choice of a specific model or preconception of how a natural surface or contour in the world behaves. The case of $c_i = 0$, like the thin-plate approximation, corresponds to an intrinsic preference for flat surfaces over curved ones, and hence a world-model composed of flat surfaces.

The optimization problem is solved in practice using conjugate gradient descent with multiple restarts to minimize the energy functional with respect to the positions of the sample points normal to the curve boundary.

3.1. *Discontinuities and Parts*

It is not merely the smoothing of the original data that we wish to achieve, but also its decomposition. This is of fundamental importance in obtaining a more concise and abstract description of the input. We do this through the introduction of surface discontinuities as part of the smoothing and interpolation process. By allowing the fitted solution to "fracture" as it is forced to fit the input data, arbitrarily close fits to the input can be achieved irrespective of λ and the curvature tuning.

As the solution to the variational problem converges to its stationary point the magnitude of the smoothness function, analogous to the potential energy density, can be measured locally. This measures how well the solution $\mathbf{u}_i(t)$ actually achieves the particular target curvature being used in the specific minimization. If the value of the smoothing term, $S(\mathbf{u}_i(t), c_i)$, is very large, it indicates that the underlying data, $\mathbf{d}(t)$ could not *naturally* be fit with a function of the desired curvature. In other words, the particular target curvature being used was not an appropriate descriptive model for the portion of the data over which the function was being fit. In this case, the fitted function must be "broken," just as a natural material would break if it was subjected to excessive stress. This leads to a piecewise evaluation of the energy functional, which can be formulated as

$$E_i(u_i, c_i) = \int_0^T \|\mathbf{u}_i(t) - \mathbf{d}(t)\|^2 + (1 - l_i(t))(\kappa_u(t) - c_i)^2 \, dt, \qquad (13)$$

where $l_i(t)$ is a *line process,* as has been applied in thin-beam smoothing [16]. The line process in a binary function that is used to specify the discontinuity loci; where $l_i(t)$ is one, the smoothing constraint is suspended at that point in that layer. A set of solutions to this problem with discontinuities is illustrated in Fig. 3.

If the constraining data has a simple crease in it, it can be modelled as a single discontinuity which can be accurately localized by a deterministic thin-plate model. When discontinuities are closely spaced, however, they interact with each other in a nonlinear manner [7, 8, 57]. A simple greedy algorithm that inserts discontinuities iteratively at locations where the surface is most highly stressed will not result in

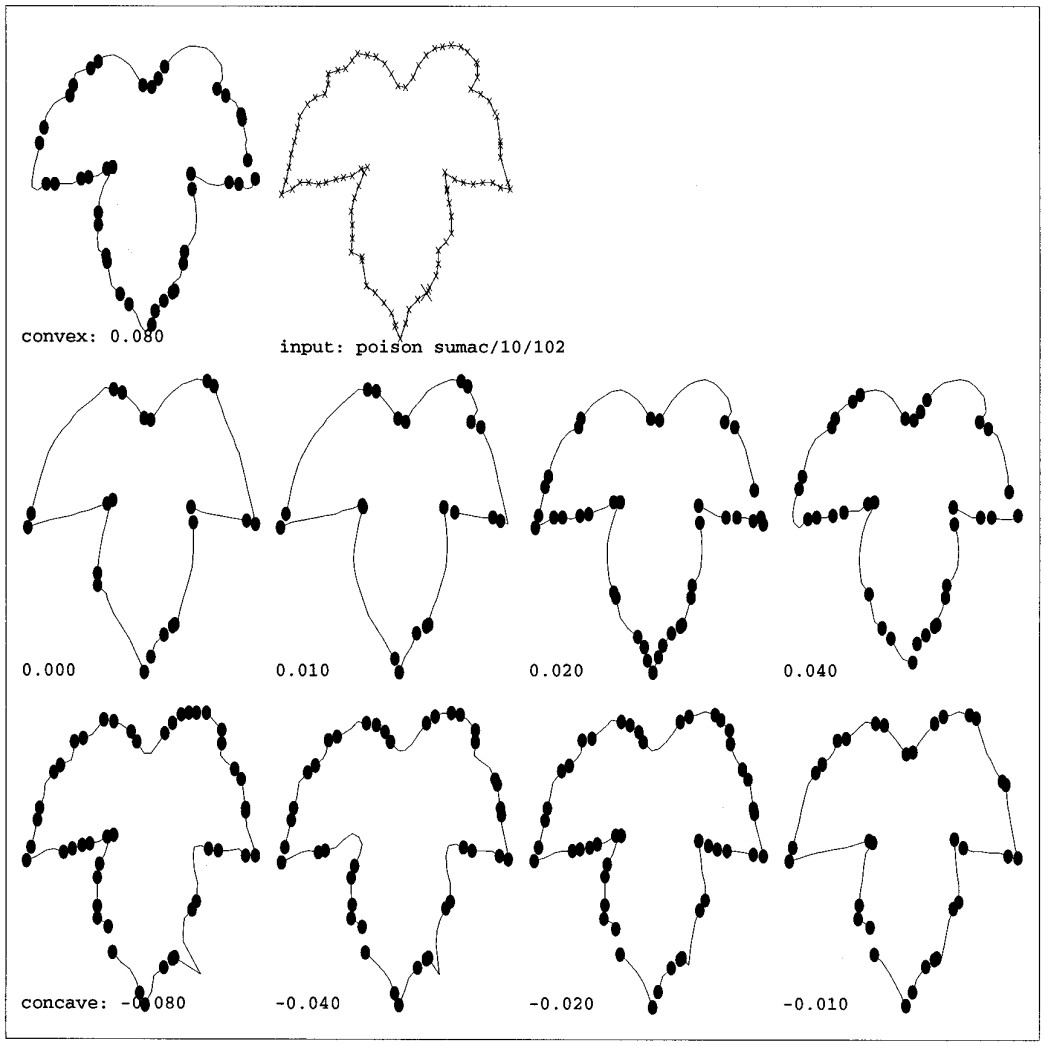

FIG. 3. A curve with discontinuities at several tunings. The poison sumac plant leaf, shown in the upper right, is processed using the curvature-tuned smoothing method using several values of the tuning parameter (before "pruning"). Small crosses on the input curve show the sample data points. The input data was actually not closed curve, but has a break in the lower right and an isolated data point away from the rest of the contour in that region. The tuning value is shown at the lower left of each decomposition. The small circles mark the discontinuities. For most tunings, the curve "shatters" into many small segments, except in regions where the tuning allows the curve to be described in a natural manner.

the optimal solution for the functional (this is illustrated in Fig. 4).

A variety of elaborate methods have been developed to insert discontinuities in an energy-minimization context [8, 16, 20, 22, 35, 40, 45, 58]. Many of these solutions could be applied in the curvature tuned smoothing context. For the application of curvature tuned smoothing described here, we require only the detection of large curve subparts or surface patches and not their precise boundaries.

Although the insertion of discontinuities is an important issue, we believe that use of sophisticated discontinuity insertion methods are not critical to the success of our method, and hence, we avoid digressing into this complex subproblem. In short, elaborate discontinuity insertion methods are needed to solve the difficult problem of optimal discontinuity detection that arises when discontinuities are closely spaced. This phenomenon is illustrated in Fig. 4.

Regions with closely spaced discontinuities necessarily are those that are not well-fit by the individual primitives under consideration. We suggest that these parts of the description (due to their intrinsic high energy) are less stable other parts of the curve (or the same parts at other scales) and, hence, are not desirable for use in recognition. As such, the problem of precisely locating these "dense" discontinuities is de-emphasised. It is for these reasons that we use a simple and efficient greedy method.

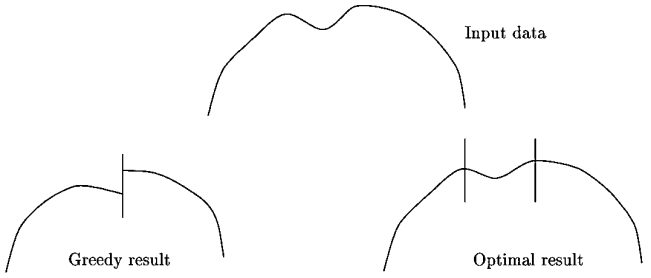

FIG. 4. Optimal discontinuity versus greedy insertion. When discontinuities are closely spaced, they interact nonlinearly and are difficult to localize. The top curve illustrates a large curve with a small valley that should (ideally) be separated by the insertion of two discontinuities. An opportunistic algorithm typically inserts a single discontinuity as shown on the left, and then is unable to insert another. The optimal solution, shown on the right, is not found.

Since the interaction between discontinuities drops off rapidly with distance (this is related to the Green's function associated with the problem, given later), we can extract discontinuities in an efficient "greedy" manner if we are willing to accept errors in precise discontinuity placement. In view of these considerations, we elect to iteratively insert discontinuities at the maxima of the energy functional. At each iteration of the optimization process, if the point of maximum energy on the curve exceeds the scale-dependent threshold ξ_i (discussed in Section 7), a discontinuity is inserted there. In practice, multiple discontinuities can be inserted simultaneously, so long as they are substantially further apart than the width of the Green's function of the minimization problem (discussed later). Inserting multiple discontinuities at once, however, is a pragmatic issue that does not alter the fundamental nature of the computation.

For regions where the data was badly fit at a particular curvature tuning and, hence, many closely space discontinuities will be introduced, the discontinuity structure may be different from the optimal arrangement [8, 14]. This is of little import since these regions are the ones that do not match the curvature at which they are being fit and hence destined to be discarded. At some other value of the curvature tuning parameter they may be represented as a smooth discontinuity-free section.

3.2. *Surfaces*

Although curvature seems to be an useful quantity for surface description as well as curves, its precise definition and application is slightly more complex. The *principal curvatures* $\kappa_1(u(x, y), x, y)$ and $\kappa_2(u(x, y), x, y)$ (denoted for simplicity as κ_1 and κ_2) completely define the local curvature structure. As with one-dimensional curvature, these quantities are invariant to rotations and translations of the surface.

For several significant classes of surface, the principal curvatures are sufficient to recover (and, hence, uniquely identify) the original surface. In particular, this is true for the important class known as graph surfaces i.e. height fields of the form $(x, y, z(x, y))$. This is the form of surface that occurs most commonly in computational vision, derived either from illumination based methods such as shape-from-shading or from range sensors[4] [6].

As in the case of one-dimensional curvature and thin rods, these surface curvatures have natural analogues in terms of the behavior of thin elastic plates. The potential energy density of a thin plate is proportional to [9, 57]

$$2H(x, y, u(x, y))^2 - K(x, y, u(x, y)), \tag{14}$$

where H and K are the mean and Gaussian surface curvatures. This is proportional to

$$\kappa_1^2 + \kappa_2^2 \tag{15}$$

which can be linearized as

$$u_{xx}^2 + u_{yy}^2 + 2u_{xy}^2 \tag{16}$$

and is known as the *quadratic variation* [19] and has been used as a stabilizer for thin plate surface modelling.

3.3. *Curvature-Tuned Smoothing on Surfaces*

Although the mean and Gaussian curvature provide the most natural analogy with plate energy, the principal curvatures are actually a more convenient computational description for generalizing curvature-tuned smoothing to surface data, due to their isomorphic computational and physical behavior. The energy density equation (Eq. (15)) for a naturally flat plate can be modified to express a thin plate with a natural curvature other than zero. In terms of the principal curvatures, this is given by

$$(\kappa_1 - c_1)^2 + (\kappa_2 - c_2)^2, \tag{17}$$

where c_1 and c_2 are the *two* target curvatures. By combining this with a constraint from the original input data, we

[4] In general, for a complete description of an arbitrary surface we require all six of the equations of Weingarten (expressed in terms of the Christoffel symbols of the second kind which, in turn, are functions of the surface partial derivatives). For the special case of compact convex surfaces, the Gaussian curvature alone is sufficient.

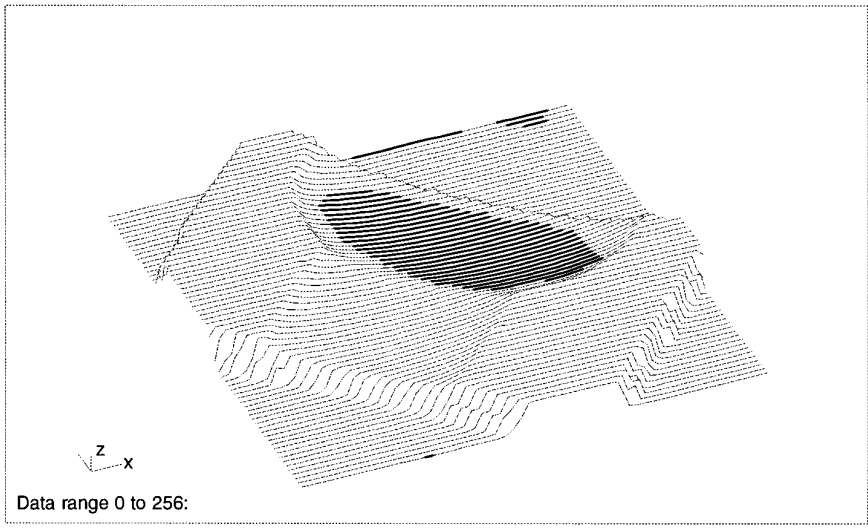

FIG. 5. Spherical region extracted from range image. The image depicts a block-shaped part with a concave depression in it. The image is from the NRC range image database. The dark lines, show the region that was extracted by a concave spherical tuning, the lighter area is labelled as poorly fit. The data was eroded to remove small ''shatter'' regions, and hence occupies only a subregion of the true spherical bowl. Note that various secondary operations such as dilation could be applied to recover more of the original area.

form the curvature-tuned smoother for surface data in three dimensions:

$$\int_S (u_i(x,y) - d(x,y))^2 + \lambda_1(\kappa_1 - c_{1,i})^2 + \lambda_2(\kappa_2 - c_{2,i})^2 \, dx \, dy. \tag{18}$$

Since the roles of κ_1 and κ_2 are physically identical, a simpler version of the surface smoother which recognizes that fact is

$$\int_S (u_i(x,y) - d(x,y))^2 + \lambda[(\kappa_1 - c_{1,i})^2 + (\kappa_2 - c_{2,i})^2] \, dx \, dy. \tag{19}$$

This leads to a two-dimensional descriptive space for object primitives. Curves are described using roughly circular regions with different curvatures. Surfaces are described with patches having different canonical two-dimensional structure: convex or concave spheroidal, cylindrical, hyperbolic, etc. A coarse sampling of this space leads to a description akin to that produced using the combinations of signs of curvature: both positive is convex spherical; one zero and one positive is convex cylindrical, etc. By sampling the two-dimensional curvature space more finely, a description with greater structural resolution is produced. That is, we can discriminate between surface regions that are highly curved cylinders and those that are gently curved cylinders. The results of the application of this process for specific tuning values are shown in Figs. 5 and 6.

4. THE INFLUENCE FUNCTION

Under appropriate conditions linear variational problems can be expressed in terms of their associated Green's function. This explicitly expresses the sensitivity of the solution at a point to perturbations in the input data at other locations. Although the nonlinear functional used for curvature-tuned smoothing does not have a Green's function *per se,* we can, nevertheless, examine its behavior in response to small input changes and, hence, assure its stability with respect to small changes in the input (hence, its conformity to Marr's principle of graceful degradation).

Furthermore, a Green's (or influence) function with local support implies that the system will have a natural and locally consistent behavior. Changes at a given location will have substantial effects primarily at its immediate neighbors. For small changes in the input, the behavior of the energy functional can be linearly approximated. Again, for simplicity, we will consider the one-dimensional case; the two-dimensional case is a simple but tedious extension. By approximating $\kappa_u(t)$ by $u''(t)$, under the assumption that $u'(t)$ remains small, we obtain what we will refer to as a *tuned version of the thin plate* model.[5] The Euler–Lagrange equation for the **"tuned thin-plate problem"**

$$E_i = \int (u_i(t) - d(t))^2 + \lambda(u_i''(t) - c_i)^2 \, dt \tag{20}$$

[5] A more general linearized version of the tuned model may be obtained by assuming that $u'(t) = g(t)$, i.e., a fixed value at each point on the function. This assumption does not alter the subsequent analysis, except to make it syntactically more unwieldy.

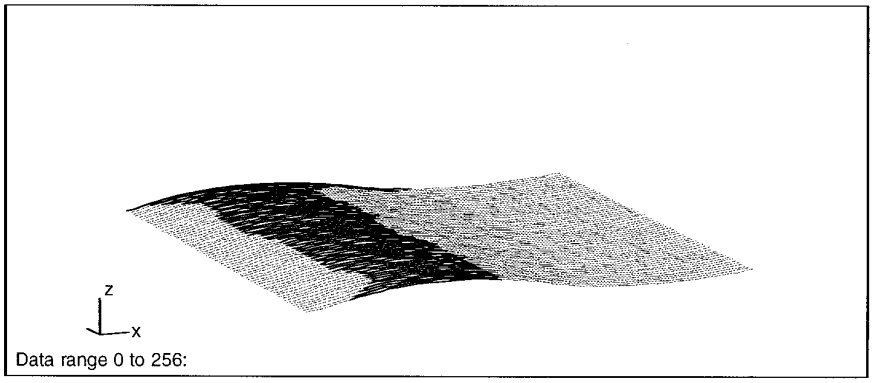

FIG. 6. Cylindrical convex region extracted from "wave" image. The image depicts a wavy object from obtained from the University of Pennsylvania GRASP lab range scanner. The dark lines show the region that was extracted by a convex tuning.

is given by

$$u_i(t) + \lambda u_i''''(t) = d(t) \tag{21}$$

independent of c_i. The natural boundary conditions for the Euler–Lagrange equation are given by[6]

$$u_i'''(0) = u_i'''(T) = 0 \tag{24}$$

and

$$u_i''(0) = u_i''(T) = c_i. \tag{25}$$

For a spatial domain of infinite extent, the Green's function is identical to that for the conventional thin-plate. The intuitive explanation for this is that for infinite extents it is impossible to bias them towards circles of substantial curvature. For an infinite line, this Green's function has the form

$$G(t_0, t) = \frac{1}{2\lambda^{1/4}} e^{-|t-t_0|/\lambda^{1/4}\sqrt{2}} \cos\left(\frac{|t-t_0|}{\lambda^{1/4}\sqrt{2}} - \pi/4\right). \tag{26}$$

[6] The boundary conditions for the full nonlinear one-dimensional CTS problem are given by

$$\left(\frac{u_i''(t)}{\sqrt{1+u_i'(t)^2}} - c\right) = 0 \tag{22}$$

and

$$2u_i'''(t)u_i'(t)^2 + u_i''(t)u_i'(t)c\sqrt{1+u_i'(t)^2} - 3u_i''(t)^2u_i'(t) + 2u_i'''(t) = 0 \tag{23}$$

(at $t = 0, T$).

The rapid decay of this function means that the effects of discontinuities or changes to the input data are high localized. It also makes explicit the role of λ as a scale constant.

5. UNIQUENESS AND LOCAL MINIMA

The desirability of a viewpoint invariant functional based on curvature, and its inherent nonlinearity, has been observed by other authors (using a form of regularizing smoothing computation different from CTS, but similar to that computed for the single layer of CTS having the tuning parameter c_i set to zero). Blake and Zisserman were among the first to comment on this ([7; 8, p. 170; 53] also make this observation and deal with it in a similar manner). They suggest solving the problem with a first stage of minimization using a membrane model to estimate the surface derivatives. A second minimization stage is then proposed, based on an approximation to curvature while using the approximate first derivatives from the first stage as if they were constants. This results in a process composed of two quadratic functionals (ignoring the discontinuity processes). Although this approach leads to a unique minimum, it does not guarantee it is the global minimum for the original problem of smoothing with minimized curvature. If the problem of smoothing with minimized curvature were indeed to be nonconvex, then the approach to solving it using a combination of two convex approximating processes would produce a consistent solution. This solution, however, would not necessarily correspond to the global minimum of the original problem.

Although the curvature-tuned smoother, even when tuned to zero curvature, does not satisfy the necessary conditions for a provably convex problem[7] [7], the exis-

[7] Convexity or local convexity follow directly if the variational problem is quadratic or the Hessian of the Euler equation is positive definite.

tence of any local minima, let alone significant ones, is not a difficulty in practice. For small values of λ the stabilizing functional $(\kappa(t) - c_i)^2$ can be approximated by $([x''(t)y'(t) - y''(t)x'(t)]/q(t) - c_i)^2$, where $q(t)$ is a constant function that approximates the first derivative, $q(t) \approx \sqrt{x'(t)^2 + y'(t)^2}$. That is, the curve's arc length remains roughly constant and the changes in the functional are manifested mainly in the more sensitive second derivative terms. When this approximation holds, the energy functional is quadratic and, hence, guaranteed to be globally convex [46].

More important, perhaps, is the fact that we are interested in solutions where $\mathbf{u}(t)$ and $\mathbf{d}(t)$ are fairly close together. The minimum for the curvature-tuned smoother is quite stable due, in part, to this condition. The behavior would seem to be characterized by the fact that *if* other local minima exist, they are far enough away from the original data not to be a problem (for example, they might be cases where the arc-length parameter becomes extremely large, hence pushing the overall curvature value to become small).

FIG. 7. Stability of solution. This figure illustrates the invariance of the final CTS solution to the initial curve estimate. The lower left shows a sparsely sampled very noisy curve. The upper left shows a perturbed starting "guess" for the solution given the same input data. The curve on the right shows the CTS "smoothed" solution with a curvature tuning of -0.02 and discontinuity insertion inhibited. This solution curve for both of these starting guesses and is essentially independent of the starting guess except for quantization effects.

5.1. *Empirical Confirmation*

To verify the predicted stable convergence of the curvature-tuned smoothing process, the sensitivity of the final solution to the initial starting data for CTS on planar curves was evaluated using randomly generated perturbations to the input. Discontinuity insertion was inhibited since the issue is the global convergence of the smoothing operator. For an ensemble of 50 input curves, alternative initial conditions for the minimization process were generated by adding random Gaussian noise to the point data defining the curves (a variety of tests based on manual deformations of the curves were also carried out). The data sets included cases with mixed convex and concave curvature as well as a fixed sign of curvature. These manually selected starting functions were then perturbed by the addition of zero mean Gaussian noise normal to the local tangent. This was carried out on between four and 10 independent trials per curve using curvature tunings in the range -0.4 and 0.4 and with standard deviations σ of the noise distribution of between one and 16 for each starting estimate.

The final solution for the maximum difference between two solutions u_1 and u_2 for any given problem data $\mathbf{d}(t)$ varied by under 1% over 50 trials. Furthermore, the bulk of this residual difference resided in the displacement of the points making up the discrete curve *tangentially* along the curve, rather than as changes in the normal direction (this is due to the particular minimization strategy used). Two examples of alternative starting point sets $u_1(t)$ and $u_2(t)$ that converged to the same final solution (using the same data set $d(t)$) are shown in Fig. 7.

Research in materials science with both numerical simulations and examination of real thin-plate solids with natural nonzero curvatures supports the supposition of the absence of problematic local minima for this type of problem [55].

6. CURVATURE SCALE SPACE

The set of decompositions defined by the *curvature tuned smoothing* operation can also be described as a scale space. The stabilizer $S(\cdot, c_i)$ used for curvature tuned smoothing has the property of selecting not only structures that can be naturally described at different curvatures, but also structures with different spatial extents (i.e., scales). This observation is important because it indicates that the representation can be used to access different aspects of an object's shape and that robustness can be obtained by using coarser scales.

The segments produced when the target curvature is close to zero must, of necessity, have low curvature. If this was not the case, the segments would have poor "smoothness" and thus have discontinuities inserted in them. These low curvature segments are approximations to circular arcs of large radius. Conversely, the segments selected when the curvature tuning is large must have high curvature. This naturally limits both their length and spatial extent.

The regularization parameter, λ, also has properties related to the extent of the influence function properties (see Eq. (26)). For this reason, it can be described as a scale parameter [8]. Since λ controls the "width" of the influence

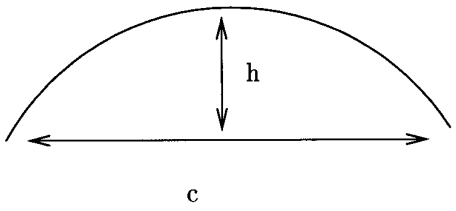

FIG. 8. Simple circle estimator.

function for the smoother,[8] it can be used to couple the influence function's spatial extent to the circumference associated with the curvature tuning. This embodies the principle that the region of support for a large circle of low curvature must be correspondingly large. This intuitive principle is borne out if we consider the estimation of the radius of a circle from a set of data points. As the radius increases, the effect of a perturbation in one of the points becomes increasingly large,

$$R(h) = \frac{c^2}{8h} + \frac{h}{2}, \tag{27}$$

where c is the chord between the two extremal points and h is the height of the perpendicular bisector (Fig. 8). For an error ε in the estimate of h, the relative error in the radius as a function of h is given by

$$\frac{R(h) - R(h+\varepsilon)}{R(h+\varepsilon)} = \frac{\varepsilon(-c^2 - 4h^2 - 4\varepsilon h)}{h(c^2 + 4h^2 + 8\varepsilon h + 4\varepsilon^2)} \tag{28}$$

which, as h becomes small (and the radius becomes large) approaches

$$\frac{R(h) - R(h+\varepsilon)}{R(h+\varepsilon)} = \frac{-\varepsilon c^2}{h(c^2 + 4\varepsilon^2)} \tag{29}$$

(or, conversely, $\partial R/\partial h = -c^2/8h^2 + 1/2$). Hence, as the radius becomes large, and h becomes small, the effect of a small perturbation can become substantial. In order to compensate for this increasing instability, a large set of data points is required. By estimating the curvature for low-curvature regions over a larger number of points the effects of small random errors is cancelled out. This relationship is exploited in adjusting the scale parameters in the next section.

[8] We can define the "width" of the influence function as the spatial range for which its envelope has some fixed fraction of its maximum amplitude, say 20% of its peak value.

7. PARAMETER INTERRELATIONSHIPS

The curvature-tuned smoothing process, as described so far, has three free parameters (for each curvature tuning layer): the curvature tuning value at which the process will be performed, the constant λ determining the amount of smoothing, and the discontinuity insertion threshold (or, alternatively, the cost of a discontinuity). Although these three parameters could be varied independently, their physical interpretations indicate that they are, in fact, coupled. The curvature tuning for the smoothing is related to scale since it specifies the curvature for the parts to be extracted during a single minimization. The stabilizing constant λ is related to spatial scale through its effect on the region of support for the local influence function. Finally, the discontinuity insertion threshold is related to the fineness of the sampling of curvature space and, hence, to the values of c_i at which the decompositions will be performed. In order for the results of the CTS process to be scale independent, we require the characteristics of the process to scale uniformly with radius of curvature.

As discussed earlier, it is appropriate for the region of support for a curvature estimate to be commensurate with the expected radius of curvature.[9]

At each layer, λ_i is chosen in proportion to the inverse of the curvature tuning since the region of support of the influence function kernel is directly proportional to the stabilizing constant:

$$\lambda_i = A/c_i. \tag{30}$$

Thus, only the constant of proportionality, A, for the entire set must be determined.

The discontinuity threshold, ξ_i, determines the precision of the segments produced at a given curvature tuning; that is, how faithful they must be to that target curvature in order to be allowable. Segments of the curve whose curvature is much higher (or lower) than the target curvature will naturally be described with lower energy or residual at the next higher (or lower) target curvature. Such segments can be broken in an attempt to find a subsection that will be well described at the current tuning. As such, the value of the discontinuity threshold is naturally associated with the distance *between* curvature tuning values. Since the curvature sampling need not be uniform, this threshold may take on different values for curvatures *greater* than the target curvature ($\kappa(t) > c_i$) and for those *less* than the target curvature ($\kappa(t) < c_i$). The discontinuity thresholds are thus given by

$$\xi_{-,i} = B|c_i - c_{i-1}| \tag{31}$$

[9] Having an influence function extent tied to the radius of curvature allows estimates to be made with uniform errors across the curvature-scale space.

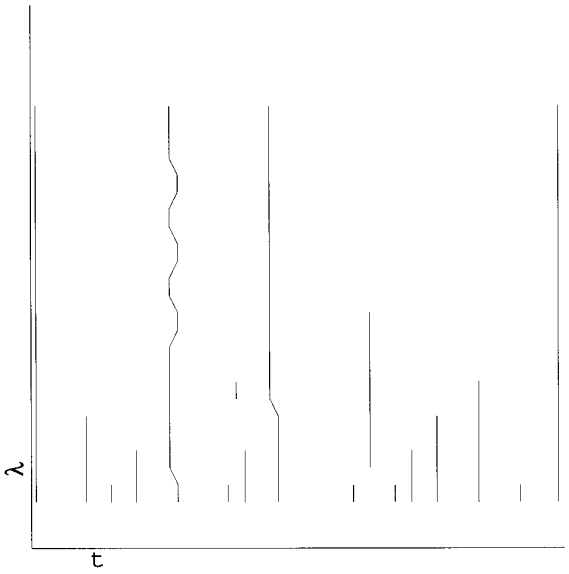

FIG. 9. Discontinuities versus tuning. The locations of discontinuities along the curve are shown for varying values of the curvature tuning. The vertical axis indicates the value of λ with zero being at the lower edge. The horizontal axis is arc-length along the curve. The lines mark the positions of discontinuities. Observe that most of the discontinuity profiles are straight and fairly continuous over substantial portions of the space. These data are extracted from the contour of the "black maple" leaf.

$$\xi_{+,i} = B|c_i - c_{i+1}|, \tag{32}$$

where B is a constant of proportionality.

7.1. *Sensitivity to Parameter Values*

A significant issue with any representation is that of its stability (or fragility) with respect to the free parameters that define it. There is no doubt that the choices of the core three parameters, λ, ξ and the set of tuning values c_i, and hence the choices of A and B, play a major role in determining the descriptions obtained from the curvature-tuned smoothing-and-decomposition process.

The choices for c_i crucially establish the class of parts that the process will extract. The discontinuity locations, and hence the description, vary only gradually with the tuning parameter. Figure 9 illustrates the incremental variations in the decomposition as c_i changes, with discontinuities shifting only slightly in position. This stable variation with c_i is important to the robust performance in the face of scale change and deformation of the data. Since the smoothness functional measures the relationship between the data and the *a priori* curvature target, the effect of scaling of the data is analogous to shifting of the tuning parameter. Sampling sufficiently to assure the description changes only slightly from one value of the tuning parameter to another, the effect of scaling is limited to a shift (translation) of the description within the tuning space. That is, segments that were represented at tunings c_a, c_b, and c_d are represented at c_{a+1}, c_{b+1}, and c_{d+1}.

This stable behavior in response to changes in the tuning parameter is a direct result of the stabilizing effect of the smoothness term. The smoothing imposed by the curvature term of the energy function reduces the potential variation of the signal's curvature, and brings it closer to the tuning value. For large values of λ the sensitivity of the discontinuity locations to the tuning value drops to zero since the curve is always forced to near-perfect smoothness. For moderate values of λ, the smoother's property of attenuating noise also attenuates variations in the signal's behavior as the tuning value changes. This is, of course, partially dependent on ξ. If ξ is small, and hence discontinuities are inserted promiscuously, then subtle variations in the solution will lead to changes in the discontinuity pattern.

8. FORMING DESCRIPTIONS

Once the series of minimizing functions and their discontinuities have been determined, a description of the input

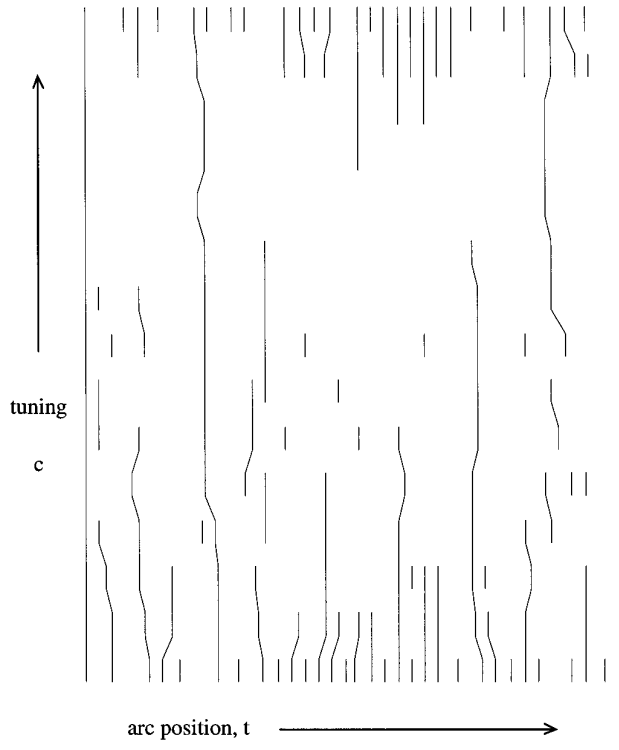

FIG. 10. Discontinuities in curvature-scale space. The locations of discontinuities in a curve are plotted as a function of position and curvature tuning. Values of the curvature tuning parameter at which they were extracted are shown along the horizontal axis, with zero being in the center. The horizontal axis is arc-length along the curve. The lines show the positions of discontinuities. Observe that most of the discontinuity profiles are straight and fairly continuous over substantial portions of the space.

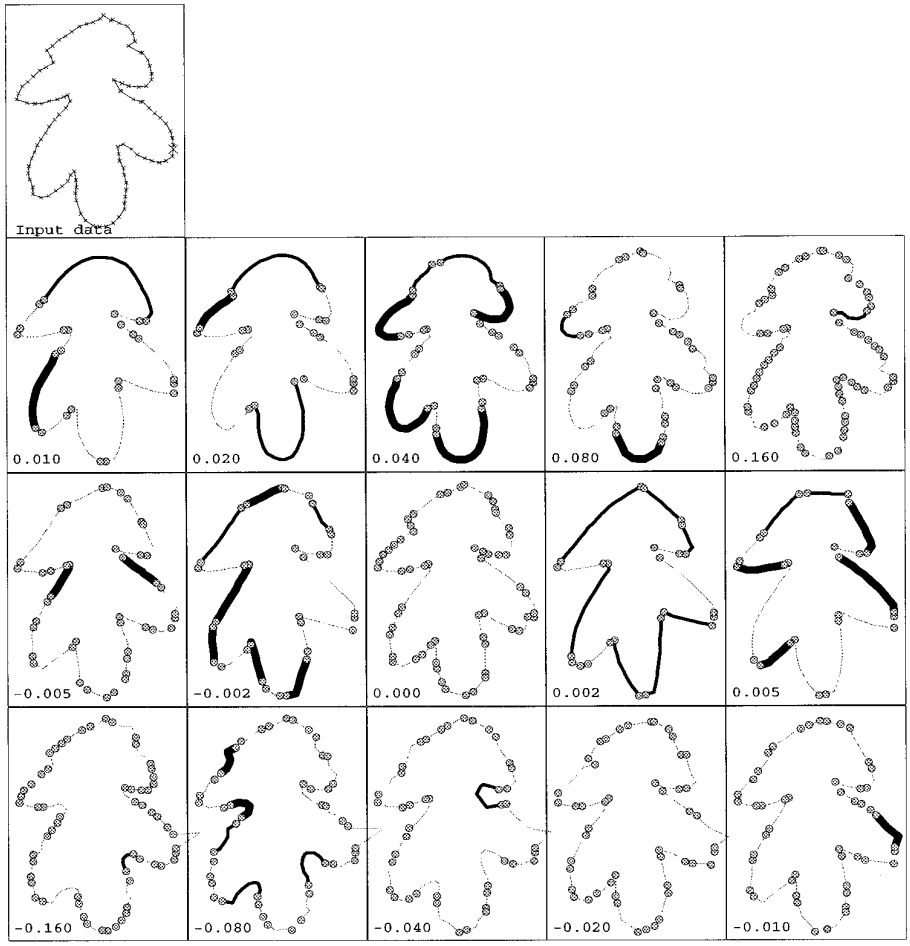

FIG. 11. Oak leaf description. An oak leaf contour described using the curvature tuned smoother. Each copy of the curve corresponds to a description using a different value of the tuning parameter. The tuning values are shown at the lower left of the descriptions. Small circles mark discontinuities. Segments extracted for the final description are drawn in heavier black. Note that for many tuning values, much of the data is not naturally described and is hence broken into many small segments.

can be obtained in terms of ''curvature-scale space.'' The locations of discontinuities (and hence of discontinuity-free regions) vary little for similar values of the tuning parameter, as with traditional thin-plate smoothing. This follows from the form of the Green's function and is illustrated Fig. 10. The locations of the boundaries of these segments tend to be stable across ''scales.'' This stability property is only natural: if we move through curvature tuning space, for a segment that can be well described using a value of the tuning parameter of c_j, it should be possible to construct a segment for tuning $c_{j+1} = c_j + \Delta c_j$ which has a similar but slightly shifted curvature and, hence, is still close to the data. Eventually, the curvature drifts so far from the natural curvature of the data that this segment ''shatters'' and the semistable description vanishes. This stability process is also consistent with the stable discontinuity locations observed for thin-plate smoothing using the GNC algorithm [8] when a discontinuity position is plotted as a function of the stabilizing constant, λ.

In order to locally evaluate the quality of the decomposition, we can determine the net value of the smoothness functional over each of the segments individually. Because the positions of the segments along the curve vary gradually as a function of the curvature tuning, the tuned smoothness $\int S(\mathbf{u}(t), t)\, dt$ of a segment can easily be compared to that of the same segment at neighboring scales except over intervals in curvature space where the segment structure changes. This allows the selection of those segments across the space of the tuning parameter that have the best (lowest) tuned-smoothness values relative to their neighbors at adjacent tunings and, hence, that serve as the most natural and consistent descriptions of the data. These are the segments from our *a priori* set of models that suit the input stimulus. By using only these segments to form our

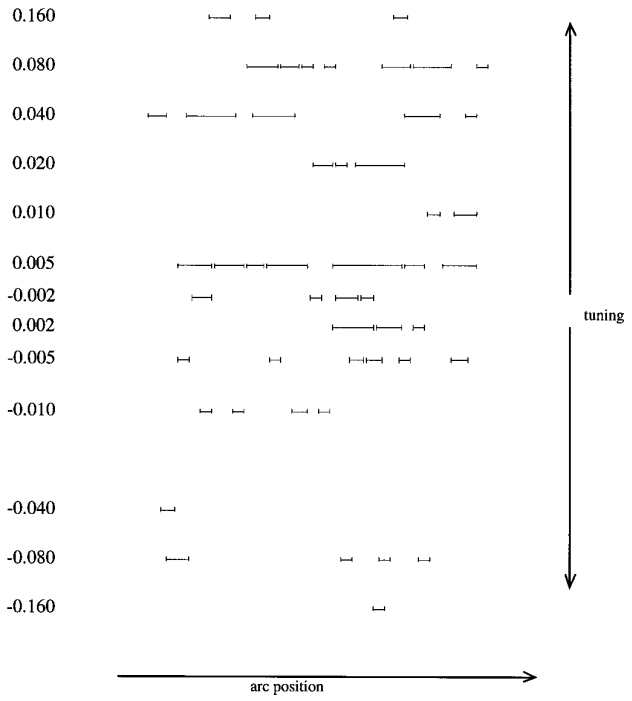

FIG. 12. Oak-leaf in curvature-scale space. Image of an oak leaf with the extracted parts shown by lines plotted with the curvature tuning at which they were extracted versus the arc-position at which they occurred.

object representation, we obtain a description that is compact in terms of the number of primitives (the segments before selection and the ensuing representation are shown in Figs. 11 and 12 while a mapping from a curve to its representation is shown in Fig. 13).

8.1. *Curve Abstraction*

Once the curve has been partitioned into segments (parts) via the insertion of discontinuities, we have imposed a large measure of structure on the original data; or conversely, we have extracted a large measure of structure. The parts themselves, being sections of roughly uniform curvature, are simple yet they carry substantial expressive power as a group. As shown earlier, a smoothly curved object such as a silhouette can be described with only some 15 or fewer segments. For simply representing categorical object structure (coarse shape) we do not need to retain all the internal point locations for each segment. The segments can be encoded in a much simpler form: by their length ($l = |t^l - t^r|$), mean position (t), and the curvature tuning used to extract them (c_i). This simple encoding will be referred to as the (type one) *segment descriptor* for a segment j,

$$s_j^1 = (t_j, l_j, c_{i,j}), \tag{33}$$

and the set of these for an object, o, constitutes its (first order) description, $S^1(o)$. Observe that the curvature tuning at which the descriptor was extracted serves as a coarse estimate of the mean curvature of the segment at a particular smoothing scale. We do not explicitly encode the curvature of the segment itself. The best-fit tuning is similar to the quantized mean curvature over the segment. This is merely an analogy, however; the tuning is not formally equivalent to the mean curvature. This estimate of mean curvature is a stable measurement; the averaged curvature over a neighborhood is a far more stable estimator than the local curvature function, even when a second-order regularizing smoother is used [24].

For smoothly curved data, this extremely simple description yields substantial disambiguating power. For data that has sharp corners, however, the angle between the corners is not likely to be *robustly* described as a smooth curve at any curvature scale. In this case, the description can be simply augmented to provide more information on the segments. This is done by adding the mean tangent information τ over the segment to the descriptor,

$$s_j^2 = (t_j, l_j, c_{i,j}, \tau_j), \tag{34}$$

to produce a slightly less compact object description, $S^2(o)$. In practice, S^1 provides great descriptive power, and is used exclusively in the matching experiments we will describe. The reason for this is simple: not only does each segment describe the data that it encompasses, but if the object is smooth and the curvature space is sampled sufficiently, there will be some segment covering every portion of the object. Hence, the orientation information *between* segments is captured implicitly by the segment that covers the region that joins them. This is illustrated in Fig. 14. If there are polygonal objects in the scene, however, it is quite likely that some vertices will have curvature so high (despite the smoothing) that no segment adequately describes them. As a consequence, the representation may provide insufficient information to determine the angle between pairs of straight edges. For example, it will not reliably permit a rectangle and a parallelogram to be distinguished. Although the rich description provided by S^1 will still be quite good, the use of S^2 may be preferable in such cases.

9. MATCHING

The segment description S^1 specified earlier can be directly used for rotation invariant object matching. This will be described below. Objects are described by a canonical set of segments (or parts), enumerated as we proceed around the contour (in practice, this listing of segments is repeated to achieve matching independent of the initial position [37]). These parts express information of varying

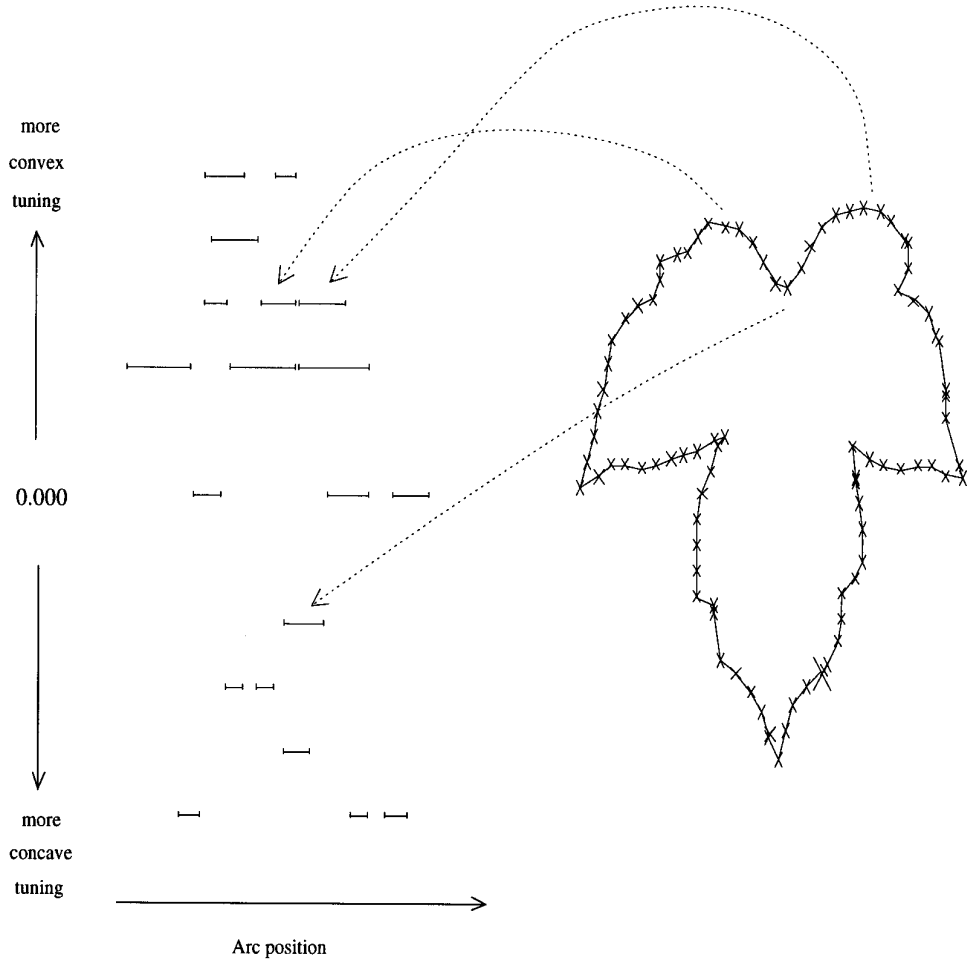

FIG. 13. Poison sumac leaf and scale-space. The CTS description of the poison sumac leaf is shown with the segments corresponding to certain features on the leaf illustrated.

degrees of locality and embody information at multiple scales at the same location. This makes the part structure a two-dimensional description: one dimension is elapsed arc length as we circumnavigate the curve; the other is curvature (or scale). Matching algorithms may exploit either one, or both of these dimensions.

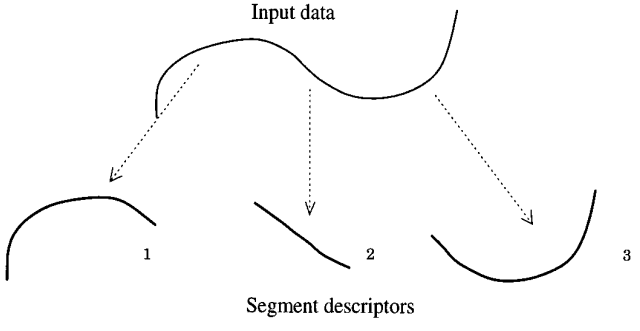

FIG. 14. Overlapping segments describing an curve. The segments describing a smooth curve overlap and hence the relative orientations between segments are coded implicitly. In this example, the relative orientation between segments 1 and 3 is constrained by segment 2.

In order to illustrate the descriptive sufficiency of the parts extracted with curvature-tuned smoothing, a specific matching method based on an even simpler description than S^1 will be constructed. This description, which we will refer to as S^0 (the type zero segment description) encodes the segment using only its ordinal position along the contour n_j (rather than the precise position), its length, and its curvature tuning:

$$s_j^0 = (n_j, l_j, c_{i,j}). \tag{35}$$

We will perform this matching using structure only in the arc-length dimension, exploiting the structure in the scale dimension only in an implicit manner. That is, a match consists of a sequence of segments along the curve being similar to a sequence from another curve. Since the similarity measure includes the curvature parameters, the scale

dimension including effects such as the containment of a small-scale segment within a coarser one (a small bump within a larger bump) is embedded in the description. The key element in the matching process is the metric for part similarity that we will define. This follows naturally from the segment descriptor S^0. To perform the matching, we can readily use a dynamic programming method [1, 17, 37]. As this method implicitly encodes the segment sequence along the contour, we will not directly encode this value in the segment comparison metric.

Segments of low curvature tend to correspond to larger spatial extents of the curve and are computed with a broader influence function. For this reason, they are more immune to noise. Furthermore, since we sample curvature space geometrically, low curvature segments are less sensitive to distortions such as those due to perspective or bending; larger absolute changes in radius are required to alter the structure. For similar reasons, segments of substantial length are also more stable than short segments.

The matching metric for type-zero segments has the form

$$\langle s_1^0, s_2^0 \rangle = |\mathrm{sgn}(c_{i,2}) \log|c_{i,1}| - \mathrm{sgn}(c_{i,2}) \log|c_{i,2}|| + |l_1 - l_2|. \tag{36}$$

Although the use of a L_1 norm for combining these parameters is not crucial, the independent combination of different types of measurement information seems perceptually appropriate. For substantially different stimulus dimensions humans also appear to use a metric close to an L_1 norm [18]. Furthermore, the L_1 norm is more efficient to compute which could be significant if the model set is large. A logarithmic weighting is applied to the curvature components to impose a resolution preference for coarse-scale information [61].

9.1. *Global Matches*

Curve matching can be formulated as a dynamic programming problem in terms of matching an increasingly long subsequence of segments from one curve to a series of segments from the other. This approach is general and can be adapted to partial matches when searching for occluded objects [1, 13, 17, 37]. It is also space efficient and avoids the use of domain dependent constructions such as hashing functions or quantization estimates for the search space.

As applied to the curve matching problem, a cost function is constructed which expresses the total cost for matching an increasingly long subsequence of segments from one curve to a series of segments from the other. The dynamic programming state number expresses the number of segments matched so far. A match is reflected by a path through the cost array corresponding to a sequence of matching segments. Invariance to the initial position on either curve is achieved by doubling the series of tokens and looking only for a substring of half the total length [13, 37]. That is, we look for the best matching of the test curve in a double-length model. By constructing a matching function that ensures that matched curves have the same *sequence* of (multiscale) primitives, matching is made robust with respect to local deformations in a curve.

This has been demonstrated using an algorithm that constructs an incrementally expanded table of costs such that for two curves composed of segments, entry $C(i, j)$ in the cost table reflects the match the first i segments that one curve makes with the first j segments from the other. The process of matching one contour with another is then a process of executing the dynamic program for an observed data set against the set of models. The complexity of the dynamic programming process for n models each comprised of up to m segments is $O(nm)$.

This procedure has been shown to be appropriate for matching curves that are noisy versions of one another or that have undergone a limited amount of deformation [13]. For pairs of curves that have significant structural variations with respect to one another, there will be substantial mismatch error. For many natural processes structural variations may be present at a global level while subparts and local structures are similar. It has been suggested that one way in which this can occur is when local generative processes at different scales are combined in a pseudo-random or nonrigid manner [61]. In such cases the alternative approach described below may be appropriate for shape recognition. The algorithm used constructs a table of costs such that for two curves composed of type-zero segments $A = s_{a,1}^0 s_{a,2}^0 s_{a,3}^0 \ldots s_{a,N}^0$ and $B = s_{b,1}^0 s_{b,2}^0 s_{b,3}^0 \ldots s_{b,N}^0$. Entry $C(i, j)$ in the cost table has the value of match the first i segments from curve A with the first j segments from curve B. The cost for the diagonal elements in the cost table is thus

$$C(i, i) = \sum_{j=0}^{i} \langle s_{a,j}^0, s_{b,j}^0 \rangle_0. \tag{37}$$

The significance of nondiagonal motions through the cost array is that they express segments from one curve that do not match a corresponding segment from the other. The cost array can thus be expressed as

$$C(i,_j) = \min \begin{cases} C(i-1, j) + \langle s_{a,i}^0, \mathrm{NONE} \rangle_0, \\ C(i, j-1) + \langle \mathrm{NONE}, s_{a,j}^0 \rangle_0, \\ C(i-1, j-1) + \langle s_{a,i}^0, s_{b,j}^0 \rangle_0. \end{cases} \tag{38}$$

The process of matching one contour with another is then a process of executing the dynamic program for an observed data set against the set of models. The complexity of the

dynamic programming process for n models each comprised of up to m segments is $O(nm)$. Since the curvature-tuned smoothing description produces a comparatively small set of segments per object this is acceptable (i.e., m is typically around 15). A simple method for accelerating the search is to use only the lowest curvature segments (i.e., the coarsest scale) for an initial matching process, followed by a second matching using the full description across curvature space against the subset of models that provided the closest matches.

10. OVERVIEW OF SYSTEM IMPLEMENTATION

In this section we review the application of the CTS representation to matching including the specification of typical values for the free parameters. The process of computing a CTS-based representation of an object and using it for matching consists of the following steps. The free parameters for the process are:

- A, the scaling factor for the stabilizing constant. In practice, a value of 1000 is used.
- B, the constant relating the discontinuity threshold to the curvature sampling rate. In practice, a value of 2 is used.

Note that the range of scales over which the computation is performed ranges from zero to the curvature $c_{\max}$ associated with the radius of a single pixel and, hence, is not a free parameter.

1. Compute the multiscale CTS representation of an unknown object. Sample curvature over a range of curvatures specified by $c_{\max}$.
2. At each scale, minimize the energy functional producing a set of segments.

(a) Compute the first CTS decomposition using $c_i = -c_{\max}$ (e.g., so that the radius of curvature matches the smallest salient feature).

(b) Compute the successive CTS decompositions using $c_i = c_{i-1}/2$ until the absolute value of c_i is close to zero (e.g., a radius of curvature is equal to the image size, in the examples shown here $c_{\min} = |c_i| = 0.005$).

(c) Compute the CTS decomposition using $c_i = c_{\min}$.

(d) Compute the successive CTS decompositions using $c_i = c_{i-1} * 2$ until the value of c_i reaches $c_{\max}$.

3. The set of descriptors for all these scales constitutes the contents of the scale space.
4. Over the resulting scale space representation, select the best descriptors (Section 8).
5. For each object in a library of known objects for which such descriptions have been computed, compare the set of descriptors using the dynamic matching method of Section 9.1.

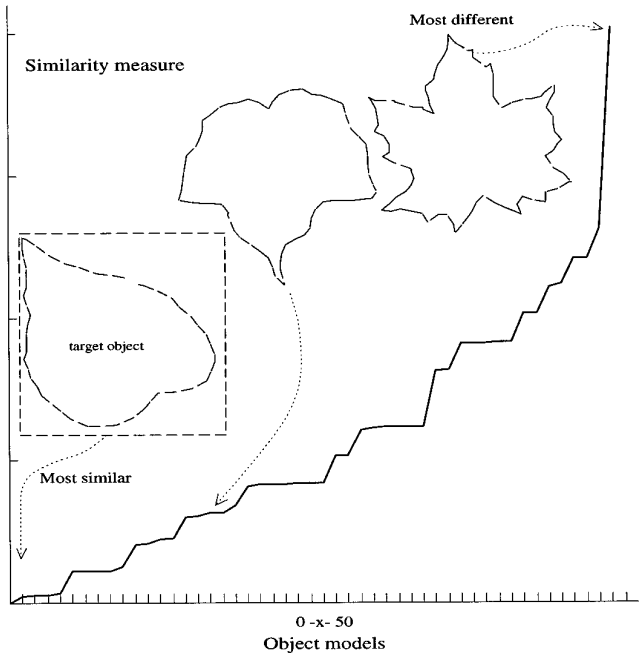

FIG. 15. Distribution of model "distances." The graph illustrates the distribution of similarity measures between different models for the segments extracted from a lilac leaf.

In experimental trials of 50 object contours were processed using the description obtained from curvature-tuned smoothing. The values of the distance metric describing the similarity of one object to another was, in general, broadly distributed among the set of objects: some objects were quite similar to the target; others varied along an axis of increasing dissimilarity. In particular, the silhouettes of leaves were consistently matched to others of the same species. Although difficult to measure objectively, this continuous similarity measure agreed well with observers' subjective appraisal of how similar the shapes were (rather than finding all nonmatching objects to be equally dissimilar). The distance distribution among possible models for one of the objects, along with some alternative models, is shown in Fig. 15.

As an application-specific modification, matching strategies such as dynamic programming can be readily tuned to look for correspondence in the scale dimension explicitly. In addition, alternative methods for matching the segments produced from curvature-tuned smoothing are possible, both for curve and surface data [11].

11. CONCLUSION

The use of a collection of curvature-based minimizing operators, which we term collectively "curvature-tuned smoothing" has been developed to address several difficulties with existing approaches to smoothing, interpola-

tion, segmentation, and curve and surface description. The new approach provides a multiscale description of objects, but one in which the notion of scale is based on curvature-scale space, rather than the conventional definition in terms of Gaussian blurring and spatial frequency. The segmentation and description provided by curvature-tuned smoothing has been shown to provide a natural and powerful vocabulary for the recognition of curved objects. The ease with which this compact yet expressive description can be applied has been demonstrated in a dynamic programming context. It shows the distinctive ability of the approach to define a broad and continuous range of similarity between objects. This allows objects to be recognized or deemed alike even when they have no identical subcontours, unlike many existing approaches to curve recognition. The recognition scheme proposed here is based on generic object recognition. In practice, it may be that recognition over a constrained set of objects can benefit from a task-specific selection of features [5]. This might be accomplished by the use of a weighting function that emphasizes specific features (i.e., in Eq. (36)).

The technique, as presented, has essentially two parameters that must be selected before it is applied. One of these is the sampling grid for curvature space, the other is a scalar value that determines the relevance of the model set. Both of these parameters may be chosen in a manner that allows effective general-purpose operation, as we have shown. On the other hand, these values may also be modified dynamically by higher-level processes allowing interesting possibilities for ''attentive'' control. For small models sets, the selection of the curvature-space sampling is straightforward. The appropriate selection of a sampling grid in general situations and the refinement of the sampling to obtain greater accuracy dynamically are questions for further investigation.

The representation computed by CTS is directly applicable to curve recognition using either the matching strategy illustrated here, or one of several possible variants. The virtues of the CTS process make the descriptions it produces suitable for other applications that are based on smoothing.

The method's shortcoming is that it is tied to a model of objects approximated as composed of locally approximately circular or quadric regions. While this appears to be appropriate for a wide range of natural objects, in particular smoothly curved ones, there are classes of objects for which this descriptive space may be inappropriate or inelegant (for example, if one wanted to model speech signals or sea urchins).

The advantages of the approach are as follows:

- The extracted descriptors include those that capture the structure of large sections of the data, as well as small sections. By virtue of this multiscale nature, noise is manifested primarily in the small-scale descriptors and a large measure of robustness is obtained.
- The method can deal with both open and closed curves and can extract the descriptive primitives from the visible portion of an occluded curve.
- The shape of many natural objects can be described in more than one way. This unavoidable ambiguity is naturally captured by the CTS method. A given region of the input can have more than one ''good'' descriptor.
- The curvature structure being extracted is not corrupted by the smoothing technique. Conventional smoothing degrades all curvature information while for CTS uniform curvature regions are stable under smoothing.
- The refined description extracted from objects whose underlying structure is smoothly curved (but noisy) is concise. Each part captures a large spatial region of the data with only a few numbers. Greater accuracy, of course, requires a more detailed description.
- The computations used for computing the description are local in nature. As a result, fast parallel implementations appear realizable [20]. This is an agreeable similarity with biological vision processes.
- The parts that are extracted, being based on curvature, are conceptually and mathematically simple and are associated with models from materials science.
- The process is tunable by higher-level processes. This can be performed dynamically.
- Combined with the matching algorithm, the experimental results show that the CTS process is effective and realistic.

Several interesting issues remain to be explored further. These include the complete development of CTS-based surface recognition, the inference of volumetric models from the curvature-based segments, and the reconstruction of the original data from the very sparse but rich CTS description.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support of Natural Sciences and Engineering Research Council and the helpful comments of Allan Jepson, Demetri Terzopoulos, and Andrew Zisserman. Avi Kak was especially helpful in refining the exposition and manuscript.

REFERENCES

1. N. Ansari and E. J. Delp, Partial shape recognition: A landmark-based approach, *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(5), 1990, 470–483.
2. H. Asada and M. Brady, The curvature primal sketch. *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(1), 1986, 2–14.
3. F. Attneave, Some informational aspects of visual perception, *Psychol. Rev.* **61,** 1954, 183–193.
4. N. Ayache and O. D. Faugeras, Hyper: A new approach for the recognition and positioning of two-dimensional objects, *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(1), 1986, 44–54.

5. J. S. Beis and D. G. Lowe, Learning indesinb functions for 3-d model-based object recognition, in *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, Seattle, Washington, June 1994,* pp. 275–280, IEEE Comput. Soc., Los Alamitos, CA, 1994.
6. P. J. Besl, *Surfaces in Range Image Understanding,* Springer-Verlag, New York, 1988.
7. A. Blake and A. Zisserman, Invariant surface reconstruction using weak continuity constraints, in *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 1986,* pp. 62–66, IEEE Comp. Soc., Los Alamitos, CA, 1986.
8. A. Blake and A. Zisserman, *Visual Reconstruction,* MIT Press, Cambridge, MA, 1987.
9. R. Courant and D. Hilbert, *Methods of Mathematical Physics, Vol. 1,* Interscience, New York, 1937.
10. M. P. DoCarmo, *Differential Geometry of Curves and Surfaces,* Prentice-Hall, Englewood Cliffs, NJ, 1976.
11. G. Dudek, Shape metrics from curvature scale-space and curvature-tuned smoothing, in *Proceedings, SPIE Conference on Geometric Methods in Computer Vision, San Diego, CA, July 1991.* Internat. Soc. Opt. Eng., Bellingham, WA, 1991.
12. G. Dudek and J. K. Tsotsos, Using curvature information in the decomposition and representation of planar curves, *NATO Advanced Study Institute of Robotics and Active Vision,* July 1989.
13. G. Dudek and J. K. Tsotsos, Shape representation and recognition from curvature, in *Proceedings, IEEE 1991 Conference on Computer Vision and Pattern Recognition, Maui, Hawaii, June 1991,* pp. 35–41.
14. G. L. Dudek, Shape representation from curvature, Ph.D. thesis, Dept. of Computer Science, University of Toronto, Toronto, Canada, December 1990.
15. G. J. Ettinger, Hierarchical object recognition using libraries of parameterized model sub-parts, AI-TR-963, MIT, 1987.
16. S. Geman and D. Geman, Stochastic relaxation, gibbs distributions and the bayesian restoration of images, *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-6(6), 1984, 721–741.
17. J. W. Gorman, O. R. Mitchell, and F. P. Kuhl, Partial shape recognition using dynamic programming, *IEEE Trans. Pattern Anal. Mach. Intell.* **10**(2), 1988, 257–266.
18. R. L. Gottwald and W. R. Garner, Filtering and condensation tasks with integral and separable dimensions, *Perception psychophys.* **18**(1), 1975, 26–28.
19. E. Grimson, *From Images to Surfaces: A Computational Study of the Human Early Visual System,* MIT press, Cambridge, MA, 1981.
20. J. G. Harris, C. Koch, E. Staats, and J. Luo, Analog hardware for detecting discontinuities in early vision, *Internat. J. Comput. Vision* **4** 1990, 211–223.
21. D. D. Hoffman and W. A. Richards, Parts of recognition. *Cognition,* **18,** 1984, 65–96.
22. J. Y. Jou and A. C. Bovik, Improved initial approximation and intensity-guided discontinuity detection in visible-surface reconstruction, *Comput. Vision Graphics Image Process.* **47,** 1989, 292–326.
23. N. Kehtarnavaz and R. J. P. deFigueiredo, A. 3-d contour segmentation scheme based on curvature and torsion, *IEEE Trans. Pattern Anal. Mach. Intell.* **10**(5), 1988.
24. D. Keren and M. Werman, Variations on regularization, in *Proceedings, 10th International Conference on Pattern Recognition, Atlantic City, NJ, June 1990,* pp. 93–97, IEEE Comput. Soc., Los Alamitos, CA, 1990.
25. B. Kimia, A. Tannenbaum, and S. W. Zucker, On the evolution of curves via a function of curvature, 1: The classical case, *J. Math. Anal. Appl.* **163**(2), 1992, 438–458.
26. B. J. Kimia and K. Siddiqi, Geometric heat equation and nonlinear diffusion of shapes and images, in *Proceedings, Conf. Comput. Vision Pattern Recognit., June 1994,* pp. 113–120, IEEE Comput. Soc., Los Alamitos, CA, 1994.
27. J. J. Koenderink and A. J. van Doorn, Photometric invariants related to solid shape, *Optica Acta* **27**(7), 1980, 981–996.
28. Y. Lamdan, J. Schartz, and H. Wolfson, On the recognition of 3-d objects from 2-d images, Robotics Research TR 122, Courant Institute, New York University, 1987.
29. A. Latto, D. Mumford, and J. Shah, 1984.
30. D. G. Lowe, *Perceptual Organization and visual Recognition,* Kluwer Academic, Boston, 1985.
31. D. G. Lowe, Organization of smooth image curves at multiple scales, in *Proceedings, IEEE 2nd International Conf. On Computer Vision, Tarpon Springs, FL, Dec. 1988.* pp. 558–567.
32. D. G. Lowe, Stabilized solution for 3-d model parameters, in *First European Conference on Computer Vision, Antibes, France, May 1990* pp. 408–412.
33. D. G. Lowe and T. O. Binford, Segmentation and aggregation: An approach to figure-ground phenomena, in *Proceedings, DARPA Image Understanding Workshop, 1983,* pp. 168–178.
34. D. H. Marimont, A representation for image curves, *AAAI,* 1984, 237–242.
35. J. L. Marroquin, S. Mitter, and T. Poggio, Probabilistic solutions of ill-posed problems in computational vision, *J. Amer. Statist. Assoc.* **82**(397), 1987, 293.
36. R. Mehrotra and W. Grosky, Shape matching utilizing indexed hypothesis generation and testing, *IEEE Trans. Rob. Automat.* **5**(1), 1989, 70–77.
37. E. Milios, Recovering shape deformation by an extended circular image representation, in *Proceedings, IEEE 2nd International Conf. on Computer Vision, Tarpon Springs, FL, Dec. 1988,* pp. 20–29.
38. F. Mokhtarian and A. Mackworth, Scale-based description and recognition of planar curves and two-dimensional shapes, *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(1), 1986, 34–43.
39. F. Mokhtarian and A. K. Mackworth, A theory of multiscale, curvature-based representation of planar curves, *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(8), 1992, 789–805.
40. D. Mumford and J. Shah, Optimal approximations by piecewise smooth functions and variational problems, *Commun. Pure Appl. Math.* **42**(5), 1988, 577–685.
41. J. Oliensis, Local reproducible smoothing without shrinkage, *IEEE Trans. Pattern Anal. Mach. Intell.* **15,** 1993, 307–312.
42. J. R. Pomerantz, L. C. Sager, and R. J. Stoever, Perception of wholes and of their component parts: some configural superiority effects, *J. Exp. Psychol.* **3**(3), 1977, 422–435.
43. J. Ponce, R. Bajcsy, D. Metaxas, T. O. Binford, D. A. Forsyth, M. Hebert, K. Ikeuchi, A. C. Kak, L. Shapiro, S. Scarloff, A. Pentland, and G. C. Stockman, Panel on object representation for object recognition, in *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, Seattle, Washington, June 1994,* pp. 147–152, IEEE Comput. Soc., Los Alamitos, CA, 1994.
44. A. P. Pridmore, J. Porrill, and J. E. W. Mayhew, Segmentation and description of binocularly viewed contours, *Image Vision Comput.* **5**(2), 1987, 132–138.
45. A. Rangarajan and R. Chellappa, Generalized graduated non-convexity algorithm for maximum a posteriori image estimation, in *Proceedings, 10th International Conference on Pattern Recognition, Vol. 2, Atlantic City, NJ,* June 1990, IEEE Comput. Soc., Los Alamitos, CA, 1990.

46. K. Rektorys, *Variational Methods in Mathematics, Science and Engineering,* Reidel, Dordrecht, 1980.

47. W. Richards and D. D. Hoffman, Codon constraints on closed 2d shapes, AI Memo 769, MIT, AI lab, May 1984.

48. P. L. Rosin, Multiscale representation and matching of curves using codons, *Graph. Models Image Process.* **55,** 1993, 286–310.

49. P. Rosin, Representing curves at their natural scales, *Pattern Recognit.* **25**(11), 1992, 1315–1325.

50. P. T. Sander and S. W. Zucker, Inferring differential structure from 3-d images: Smooth cross sections of fiber bundles, TR-CIM-88-6, McGill University, 1988.

51. E. Saund, *The Role of Knowledge in Visual shape Representation,* Ph.D. thesis, MIT AI lab, Oct. 1988.

52. E. Saund, Adding scale to the primal sketch, in *Proc. of the Conf. on Computer Vision and Pattern Recognition, San Diego, CA, June 1989,* IEEE Comput. Soc., Los Alamitos, CA, 1989.

53. R. L. Stevenson and E. J. Delp, Invariant reconstruction of visual surfaces, in *Proceedings, IEEE Workshop on Interpretation of 3-D Scenes, Austin, TX, November 1989,* IEEE Comput. Soc., Los Alamitos, CA, 1990.

54. J. J. Stoker, *Nonlinear Elasticity,* Gordon & Breach, New York, 1968.

55. T. R. Tauchert and W. Y. Lu, Large deformation and postbuckling behavior of an initially deformed rod, *Int. J. of Non-Linear Mechanics,* **22**(6), 1987, 511–520.

56. D. Terzopoulos, The role of constraints and discontinuities in visible surface reconstruction, in *Proceedings, International Joint Conference of Artificial Intelligence, Karlsruhe, West Germany, 1983.*

57. D. Terzopoulos, *Multiresolution computation of visible-surface representations,* Ph.D., Dept. EE and CS, Massachusetts Institute of Technology, January 1984.

58. D. Terzopoulos, Regularization of inverse visual problems involving discontinuities, *IEEE Trans. Pattern Analysis and Machine Intelligence,* **8**(4), 1986, 413–424.

59. J. K. Tsotsos, Knowledge and the visual process: Content, form and use, *Pattern Recognit.* **17**(1), 1984, 13–27.

60. N. Ueda and S. Suzuki, Automatic shape model acquisition using multiscale segment matching, in *Proceedings, 10th International Conference on Pattern Recognition, Atlantic City, NJ, June 1990,* pp. 897–902, IEEE Comput. Soc., Los Alamitos, CA, 1990.

61. A. P. Witkin, Scale-space filtering, in *Proceedings, 3rd Internat. Joint Conf. on Artificial Intelligence, Vol. 2, Karlsruhe, West Germany, 1983.*

62. L. B. Wolff and J. Fan, Segmentation of surface curvature using a photometric invariant, in *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, June 1994,* pp. 23–30, IEEE Comput. Soc., Los Alamitos, CA, 1994.

63. M. Worring and A. Smeulders, Digital curvature estimation, *Comput. Vision Graphics Image Process. Image Understanding* **58**(3), 1993, 366–382.

64. A. Yuille, Energy functions for early vision and analog networks, *Biol. Cybern.* **61,** 1989, 115–123.

65. A. L. Yuille, Zero crossings on lines of curvature, *Comput. Vision Graphics Image Process.* **45,** 1989, 68–87.

66. P. Zhu and P. M. Chirlian, On critical point detection of digital shapes, *IEEE Trans. Pattern Anal. Mach. Intell.* **17**(8), 1995, 737–748.

67. S. W. Zucker, C. David, A. Dobbins, and L. Iverson, The organization of curve detection: Coarse tangent fields and fine spline coverings, in *Proceedings, IEEE 2nd International Conf. on Computer Vision, Tarpon Springs, FL, Dec. 1988,* pp. 568–577.