A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves

Farzin Mokhtarian and Alan K. Mackworth

Abstract— **This paper presents a multiscale, curvature-based shape representation technique for planar curves that satisfies several criteria that are considered necessary for general-purpose shape representation methods. As a result, the representation is suitable for tasks that call for recognition of a noisy curve of arbitrary shape at an arbitrary scale or orientation.**

The method rests on the concept of describing a curve at varying levels of detail using features that are invariant with respect to transformations that do not change the shape of the curve. Three different ways of computing the representation are described in this paper. These three methods result in three different representations: the curvature scale space image, the renormalized curvature scale space image, and the resampled curvature scale space image.

The process of describing a curve at increasing levels of abstraction is referred to as the *evolution* or *arc length evolution* of that curve. Several evolution and arc length evolution properties of planar curves are described in this paper. Some of these results show that evolution and arc length evolution do not change the physical interpretation of planar curves as object boundaries, and some characterize possible behaviors of planar curves during evolution and arc length evolution. Others impose constraints on the location of a planar curve as it evolves. Together, these results provide a sound theoretical foundation for the representation methods introduced in this paper.

Index Terms— **Arc length evolution, arc length parametrization, curvature scale space, curvature zero crossings, evolution, multiscale representations, planar curves, shape representation and criteria.**

I. Introduction

THIS paper introduces a novel theory of multiscale, curvature-based shape representation for planar curves. It should be pointed out that only the problem of representing the shape of a planar curve that has been extracted from an image or input by a user has been addressed in this paper. We believe the problem of extracting such a curve from an image (the *segmentation problem*) is, in general, a separate problem and should not necessarily be considered to be part of the shape representation problem [34]. We also believe that the segmentation problem can be addressed effectively by making use of knowledge of the image and scene under consideration. For example, we made use of *a priori* knowledge of band 7 Landsat images of land/water scenes to arrive at a good segmentation of such an image [25]. The boundary curves thus obtained were then matched to curves from a map of the same area using their curvature scale space representations. As a result, the correct registration between the Landsat image and the map was computed.

Some of the results mentioned in this paper have been presented elsewhere, but this paper is our first comprehensive treatment of multiscale representation techniques.

A useful general-purpose shape representation method in computational vision should make accurate and reliable recognition of an object possible. Therefore, such a representation should necessarily satisfy a number of criteria. The following is a list of such criteria. Note that when two planar curves are described as having the *same shape*, there exists a transformation consisting of uniform scaling, rotation, and translation, which will cause one of those curves to overlap the other.

Invariance: If two curves have the same shape, they should also have the same representation.

Uniqueness: If two curves do not have the same shape, they should have different representations.

Stability: If two curves have a small shape difference, their representations should also have a small difference, and if two representations have a small difference, the curves they represent should also have a small shape difference.

The importance of the invariance criterion is that it guarantees that all curves with the same shape will have the same representation. It will therefore be possible to conclude that two curves have different shapes by observing that they have different representations. Without the invariance criterion, two curves with the same shape may have different representations.

The uniqueness criterion is important since it guarantees that two curves with different shapes will have different representations. It will therefore be possible to conclude that two curves have the same shape by observing that they have the same representation. Without the uniqueness criterion, two curves with different shapes may have the same representation.

The significance of the stability criterion is that it guarantees that a small change in the shape of a curve will not cause a large change in its representation, and a small difference between two representations does not indicate a large shape difference between the curves they represent. As a result, when two representations are close, the curves they represent are close in shape, and when two representations are not close, the curves they represent are not close in shape. When this criterion is satisfied, the representation can be considered to

Manuscript received September 25, 1989; revised November 15, 1990. This work was supported by the Natural Sciences and Engineering Research Council of Canada, the Canadian Institute for Advanced Research, the Institute for Robotics and Intelligent Systems, and the University of British Columbia. Recommended for acceptance by Associate Editor C. Dyer.

F. Mokhtarian is with NTT Basic Research Laboratories, Information Science Research Laboratory, Tokyo, Japan.

A. K. Mackworth is with the Department of Computer Science, University of British Columbia, Vancouver, B.C., Canada V6T 1W5.

IEEE Log Number 9200104.

be stable with respect to noise. One way to measure the shape difference between two planar curves is the *Hausdorf distance* [12].

It is useful for a shape representation to satisfy a number of additional properties in order to become suitable for practical shape recognition tasks in computer vision. The following is a list of such criteria. Note that similar criteria have been proposed in [28], [25] and [11].

Efficiency: The representation should be efficient to compute and store. This is important since it may be necessary for an object recognition system to perform real-time recognition. By *efficient*, we mean that the computational complexity should be a low-order polynomial in time and space (and in the number of processors if a parallel computing architecture is used) as a function of the size of the input curve.

Ease of implementation: If two or more competing representations exist, it is advantageous to choose one of those representations such that the implementation of the computer program that computes that representation requires the least time spent on programming and debugging.

Computation of shape properties: It may be useful to be able to determine properties of the shape of a curve using its representation. For example, if a curve has a symmetric shape, it may be desirable to be able to determine that fact from its representation (the *symmetry* criterion). Furthermore, if the shape of a whole curve or part of a curve is the same as the shape of part of another curve, it may be useful to be able to determine that relationship using their representations (the *part/whole* criterion).

Shape representation methods for planar curves previously proposed in computational vision fail to satisfy one or more of the criteria outlined above. Note, however, that each may be quite suitable for special-purpose shape representation and recognition tasks. The *Hough transform* has been used to detect lines [14], circles [7], and arbitrary shapes [2]. Edge elements in the image vote for the parameters of the objects of which they are parts. The votes are accumulated in a parameter space. The peaks of the parameter space then indicate the parameters of the objects searched for. *Chain encoding* [8], [24] techniques approximate a curve using line segments lying on a grid. *Polygonal approximations* [29]–[31] of a curve are computed by using various criteria to determine "breakpoints" that yield the "best" polygon. The *medial axis transform* [4], [21] computes the skeleton of a 2-D object by a thinning algorithm that preserves region connectivity. *Shape factors and quantitative measurements* [5] use one or more global quantitative measurements of the object such as area, perimeter, and compactness as a description of its shape. *Strip trees* [6], [2] are a set of approximating polygons ordered such that each polygon approximates the curve with less approximation error than the previous polygon. *Splines* [3] represent a curve using a set of analytic and smooth curves. The *smoothing splines* [35] method parametrizes the curve to obtain two coordinate functions. Cross-validated regularization [37] is then used to arrive at an "optimal" smoothing of each coordinate function. The smoothed functions together define a new smooth curve. *Fourier descriptors* [32] represent a curve by the coefficients of the Fourier expansion of a parametric representation of the curve. The *curvature primal sketch* [1] technique approximates the curve using a library of analytic curves. Then, the curvature function of the approximating curve is computed and convolved with a Gaussian of varying standard deviation. The *extended circular image* [13] is the 2-D equivalent of the *extended Gaussian image*. In the extended circular image, one is given the radius of curvature as a function of normal direction. *Volumetric diffusion* [18] defines a geometrical object by way of its "characteristic function" $\chi(\boldsymbol{r})$, which equals unity when the point $\boldsymbol{r}$ belongs to the object and zero otherwise. The object is then blurred by requiring that its characteristic function satisfy the diffusion equation. The boundary of each blurred object is defined by the equation $\chi(\boldsymbol{r}) = 0.5$ or by applying the Laplacian operator to the blurred function. Richards *et al.* [33] located curvature extrema on a 2-D contour at multiple scales. Those extrema were then used to encode the contour shape.

A multiscale representation for 1-D functions was first proposed by Stansfield [36] and later developed by Witkin [38]. The function $f(x)$ is convolved with a Gaussian function as its variance σ^2 varies from a small to a large value. The zero crossings of the second derivative of each convolved function are extracted and marked in the $x - \sigma$ plane. The result is the *scale space image* of the function.

The *curvature scale space image* was introduced by Mokhtarian and Mackworth [25] as a new shape representation for planar curves. The representation is computed by convolving a path-based parametric representation of the curve with a Gaussian function, as the standard deviation of the Gaussian varies from a small to a large value, and extracting the curvature zero-crossing points of the resulting curves. The representation is essentially invariant under rotation, uniform scaling, and translation of the curve. This and a number of other properties makes it suitable for recognizing a noisy curve of arbitrary shape at any scale or orientation. The process of describing a curve at increasing levels of abstraction is referred to as the *evolution* of that curve. The evolution of a planar curve and the curvature scale space image are described in in Section II.

Mackworth and Mokhtarian [20] introduced a modification of the curvature scale space image referred to as the *renormalized curvature scale space image*. This representation is computed in a similar fashion, but the curve is reparametrized by arc length after convolution. As was demonstrated in [20], the renormalized curvature scale space image is more suitable for recognizing a curve with nonuniform noise added to it. Section III contains a brief description of the renormalized curvature scale space image.

The *resampled curvature scale space image* is a substantial refinement of the curvature scale space based on the concept of *arc length evolution*. It is shown that the resampled curvature scale space image is more suitable than the renormalized curvature scale space image for recognition of curves with added nonuniform noise or when local shape differences exist. The arc length evolution of a planar curve and the resampled curvature scale space image are described in detail in Section IV.

Section V contains descriptions of the evolution and arc length evolution properties of planar curves and discusses the significance of each of those properties. Almost all these properties are shown to be true of both evolution and arc length evolution. Together, these properties provide a theoretical foundation for the representation methods proposed in this paper. The proofs of the theorems of Section V that have not appeared before are given in the Appendix.

Section VI presents an additional experiment carried out to demonstrate the stability of the curvature scale space image with respect to noise. It also presents a table comparing the representations introduced here and an evaluation of those representations according to the criteria of Section I.

Section VII presents the conclusions of this paper.

II. The Curvature Scale Space Image

A planar curve is a set of points whose position vectors are the values of a continuous, vector-valued function. It can be represented by the parametric vector equation

$$\boldsymbol{r}(u) = (x(u), y(u)). \tag{2.1}$$

The function $\boldsymbol{r}(u)$ is a parametric representation of the curve. A planar curve has an infinite number of distinct parametric representations. A parametric representation in which the parameter is the arc length s is called a *natural* parametrization of the curve. A natural parametrization can be computed from an arbitrary parametrization using the following equation:

$$s = \int_0^u |\dot{\boldsymbol{r}}(v)| dv.$$

where $\cdot$ represents the derivative, i.e., $\dot{\boldsymbol{r}} = d\boldsymbol{r}/dv$. For any parametrization

$$\dot{\boldsymbol{r}}(u) = (\dot{x}(u), \dot{y}(u))$$

$$|\dot{\boldsymbol{r}}(u)| = (\dot{x}^2 + \dot{y}^2)^{1/2}$$

$$\boldsymbol{t}(u) = \frac{\dot{\boldsymbol{r}}}{|\dot{\boldsymbol{r}}|} = \left(\frac{\dot{x}}{(\dot{x}^2 + \dot{y}^2)^{1/2}}, \frac{\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{1/2}} \right)$$

$$\boldsymbol{n}(u) = \left(\frac{-\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{1/2}}, \frac{\dot{x}}{(\dot{x}^2 + \dot{y}^2)^{1/2}} \right)$$

where $\boldsymbol{t}(u)$ and $\boldsymbol{n}(u)$ are the tangent and normal vectors at u, respectively. For any planar curve, the vectors $\boldsymbol{t}(u)$ and $\boldsymbol{n}(u)$ must satisfy the simplified Serret-Frenet vector equations [10]:

$$\dot{\boldsymbol{t}}(s) = \kappa(s)\boldsymbol{n}(s) \qquad \dot{\boldsymbol{n}}(s) = -\kappa(s)\boldsymbol{t}(s)$$

where $\kappa(s)$ is the curvature of the curve at s and is defined as

$$\kappa(s) = \lim_{h \to 0} \frac{\phi}{h}$$

where ϕ is the angle between $\boldsymbol{t}(s)$ and $\boldsymbol{t}(s+h)$. Now, observe that

$$\dot{\boldsymbol{t}}(s) = \frac{d\boldsymbol{t}}{ds} = \frac{d\boldsymbol{t}}{du}\frac{du}{ds}.$$

Therefore

$$\frac{d\boldsymbol{t}}{du} = \frac{ds}{du}\kappa\boldsymbol{n} = |\dot{\boldsymbol{r}}|\kappa\boldsymbol{n}.$$

Hence

$$\kappa(u) = \frac{\dot{\boldsymbol{t}}.\boldsymbol{n}}{|\dot{\boldsymbol{r}}|}.$$

Differentiating the expression for $\boldsymbol{t}(u)$, we obtain

$$\dot{\boldsymbol{t}}(u) = \left(\frac{-\dot{y}(\dot{x}\ddot{y} - \ddot{x}\dot{y})}{(\dot{x}^2 + \dot{y}^2)^{3/2}}, \frac{\dot{x}(\dot{x}\ddot{y} - \ddot{x}\dot{y})}{(\dot{x}^2 + \dot{y}^2)^{3/2}} \right).$$

It now follows that

$$\kappa(u) = \frac{\dot{x}(u)\ddot{y}(u) - \dot{y}(u)\ddot{x}(u)}{(\dot{x}(u)^2 + \dot{y}(u)^2)^{3/2}}.$$

Therefore, it is possible to compute the curvature of a planar curve from its parametric representation. Special cases of the parametrization yield simplifications of these formulas. If w is the normalized arc length parameter, then

$$k(w) = \dot{x}(w)\ddot{y}(w) - \ddot{x}(w)\dot{y}(w).$$

Given a planar curve

$$\Gamma = \{(x(w), y(w)) | w \in [0,1]\}$$

where w is the normalized arc length parameter, an *evolved* version of that curve is defined by

$$\Gamma_\sigma = \{(X(u,\sigma), Y(u,\sigma)) | u \in [0,1]\}$$

where

$$X(u,\sigma) = x(u) \circledast g(u,\sigma) \qquad Y(u,\sigma) = y(u) \circledast g(u,\sigma).$$

$g(u,\sigma)$ denotes a Gaussian of width σ [22] defined by

$$g(u,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-u^2}{2\sigma^2}}.$$

Functions $X(u,\sigma)$ and $Y(u,\sigma)$ are given explicitly by

$$X(u,\sigma) = \int_{-\infty}^{\infty} x(v) \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(u-v)^2}{2\sigma^2}} dv$$

$$Y(u,\sigma) = \int_{-\infty}^{\infty} y(v) \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(u-v)^2}{2\sigma^2}} dv.$$

The curvature of Γ_σ is given by

$$\kappa(u,\sigma) = \frac{X_u(u,\sigma)Y_{uu}(u,\sigma) - X_{uu}(u,\sigma)Y_u(u,\sigma)}{(X_u(u,\sigma)^2 + Y_u(u,\sigma)^2)^{3/2}}$$

where

$$X_u(u,\sigma) = \frac{\partial}{\partial u}(x(u) \circledast g(u,\sigma)) = x(u) \circledast g_u(u,\sigma)$$

$$X_{uu}(u,\sigma) = \frac{\partial^2}{\partial u^2}(x(u) \circledast g(u,\sigma)) = x(u) \circledast g_{uu}(u,\sigma)$$

$$Y_u(u,\sigma) = y(u) \circledast g_u(u,\sigma)$$

$$Y_{uu}(u,\sigma) = y(u) \circledast g_{uu}(u,\sigma).$$

The process of generating the ordered sequence of curves $\{\Gamma_\sigma \mid \sigma \geq 0\}$ is referred to as the *evolution* of Γ.

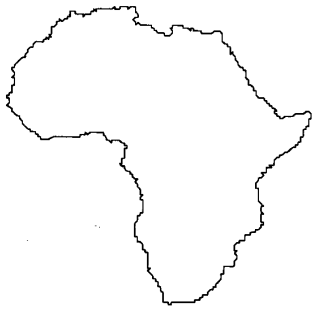

Fig. 1. Shoreline of Africa.

Fig. 1 shows a planar curve depicting the shoreline of Africa. Fig. 2 shows several evolved versions of that curve for increasing values of σ. Note that when a planar curve evolves according to the process defined above, its total arc length shrinks. The amount of shrinkage is directly proportional to the value of σ. In certain applications, this may be an undesirable feature. For example, the evolution process defined above may be used to smooth edges extracted from an image by an edge detector. However, it may be advantageous to have the smoothed edges as close as possible to the physical location of the original edges. This can be accomplished by estimating the amount of movement at each point on the smoothed edges and adding a vector to the location vector of that point to compensate for that movement [19]. The result is a smoothed curve that is physically closer to the original curve.

The function defined implicitly by

$$\kappa(u, \sigma) = 0$$

is the *curvature scale space image* of Γ [25]. Fig. 3 shows the curvature scale space of the curve of Fig. 1. Horizontal lines have been drawn across that image to indicate the values of σ that were used to compute the evolved curves of Fig. 2. Fig. 4 shows Koch's snowflake curve and several of its evolved versions. Fig. 5 shows the curvature scale space image of the snowflake curve with horizontal lines across the image to show the values of σ used in Fig. 4. Fig. 6 shows a design from a Persian carpet and several evolved versions. Fig. 7 shows the curvature scale space image of that design. Again, horizontal lines across the image show the values of σ used in Fig. 6.

We then carried out two experiments to test the stability of the curvature scale space image under conditions of noise. Fig. 8 shows the coastline of Africa with a significant amount of uniform, random noise added to it. Fig. 9 shows the curvature scale space image of Africa with uniform noise superimposed on the curvature scale space image of Africa. As expected, the images shown in Fig. 9 show differences in detail. However, a remarkable similarity in the basic structures of the two images can be observed. This experiment shows that the curvature scale space image is very reliable and stable even when a significant amount of uniform noise corrupts the shape of the input curve. The next experiment tested the behavior of the curvature scale space image under severe noise conditions. Fig. 10 shows the coastline of Africa with severe, uniform noise

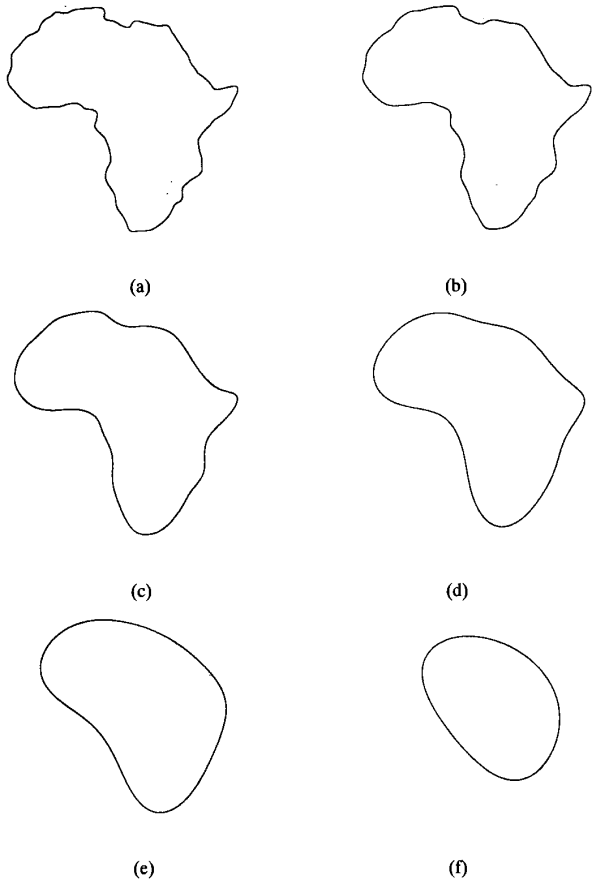

Fig. 2. Africa during evolution: (a) $\sigma = 2$; (b) $\sigma = 4$; (c) $\sigma = 8$; (d) $\sigma = 16$; (e) $\sigma = 32$; (f) $\sigma = 64$.

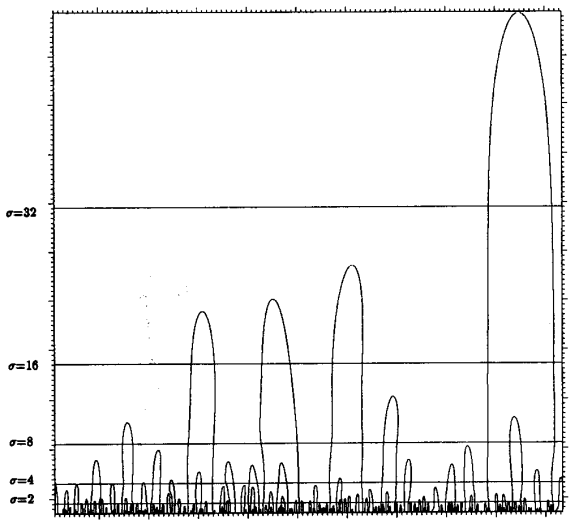

Fig. 3. Curvature scale space image of Africa.

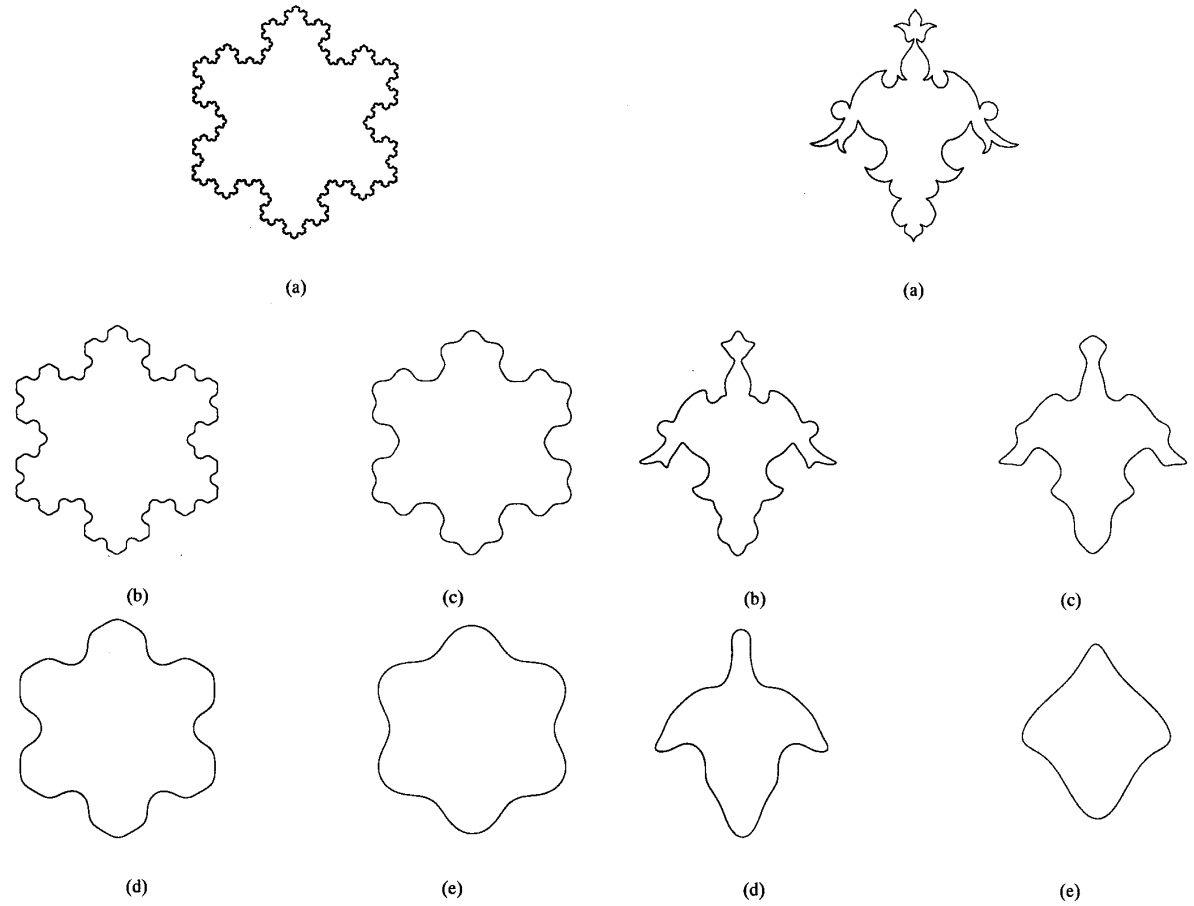

Fig. 4. Koch's snowflake during evolution: (a) Koch's snowflake; (b) $\sigma = 2$; (c) $\sigma = 5$; (d) $\sigma = 10$; (e) $\sigma = 20$.

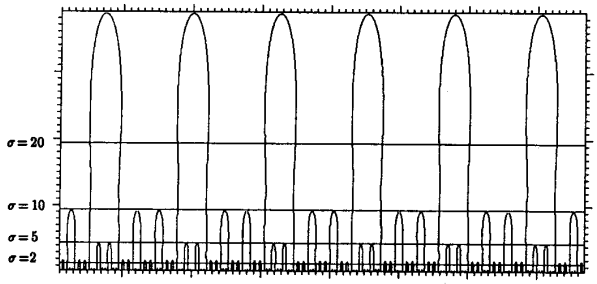

Fig. 5. Curvature scale space image of the snowflake.

Fig. 6. Carpet design during evolution: (a) Design from a Persian carpet; (b) $\sigma = 5$; (c) $\sigma = 10$; (d) $\sigma = 20$; (e) $\sigma = 50$.

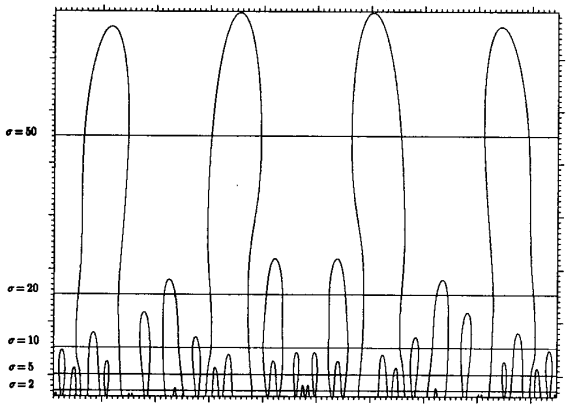

Fig. 7. Curvature scale space image of the carpet design.

added to it. Fig. 11 shows the curvature scale space image of Africa with severe noise superimposed on the it. Even with the presence of severe noise, a very close similarity can be observed between the two images.

III. The Renormalized Curvature Scale Space Image

Mackworth and Mokhtarian [20] observed that although w is the normalized arc length parameter on the original curve Γ, the parameter u is *not*, in general, the normalized arc length parameter on the evolved curve Γ_σ. Fig. 12 shows the shoreline of Africa with noise added to its lower half. Fig. 13 shows the curvature scale space of that curve. A comparison of Figs. 3 and 13 shows that a good match of one curvature scale space image to the other does not exist. To overcome this problem, Mackworth and Mokhtarian [20]

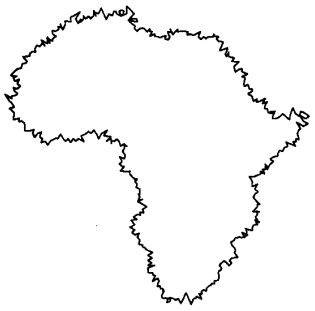

Fig. 8. Africa with uniform noise.

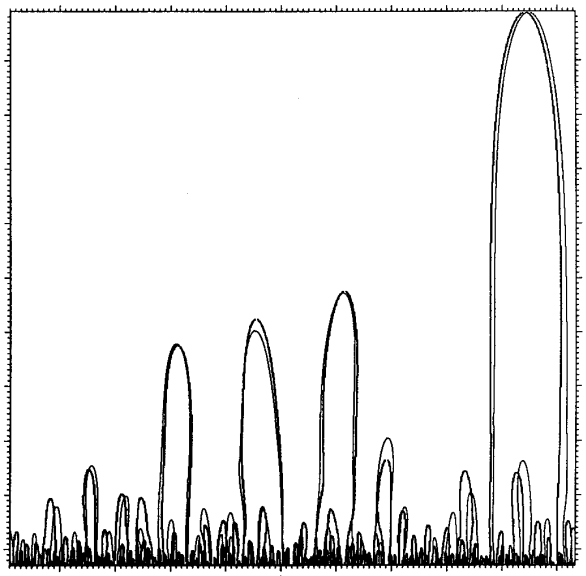

Fig. 9. Superposition of the curvature scale space image of Africa and the curvature scale space image of Africa with uniform noise.

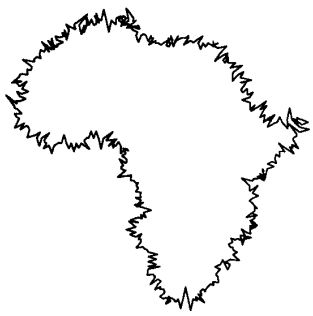

Fig. 10. Africa with severe uniform noise.

proposed the *renormalized* curvature scale space image. Let $\boldsymbol{R}(u,\sigma) = (X(u,\sigma), Y(u,\sigma))$ and

$$w = \Phi_\sigma(u) = \frac{\int_0^u |\boldsymbol{R}_v(v,\sigma)|dv}{\int_0^1 |\boldsymbol{R}_v(v,\sigma)|dv}.$$

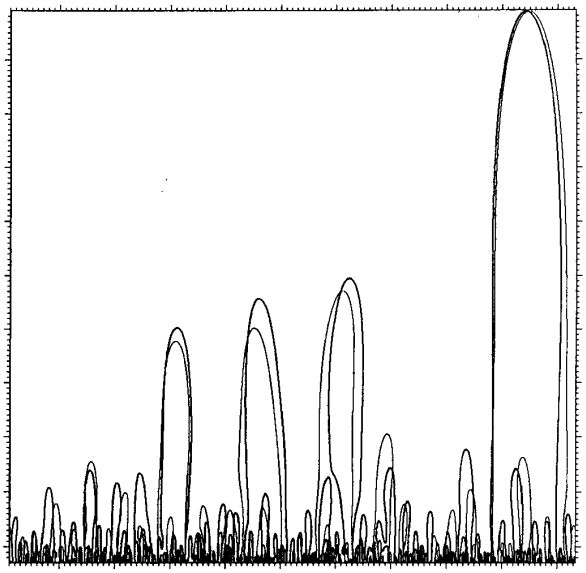

Fig. 11. Superposition of the curvature scale space image of Africa and the curvature scale space image of Africa with severe uniform noise.

Fig. 12. Africa with nonuniform noise.

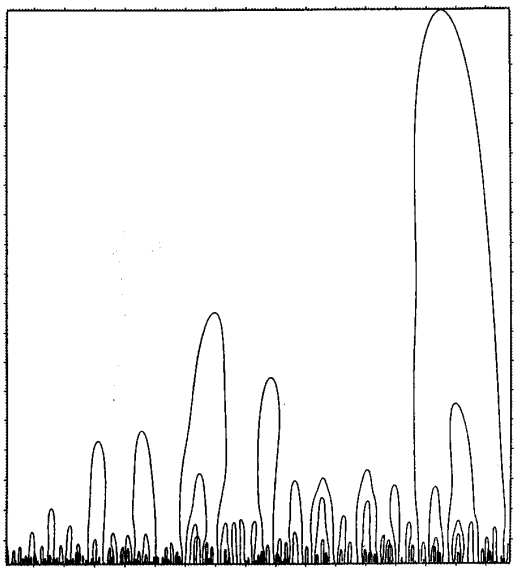

Fig. 13. CSS of Africa with nonuniform noise.

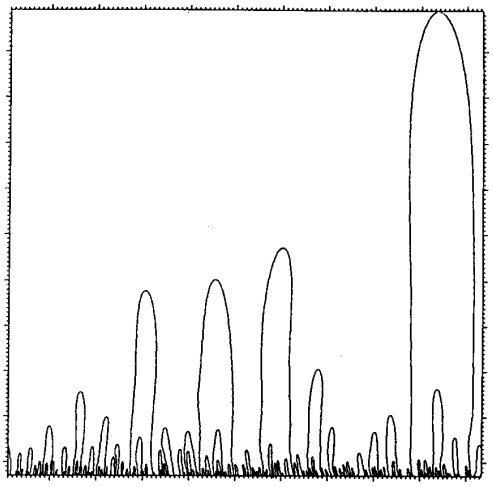

Fig. 14. Renormalized CSS of Africa.

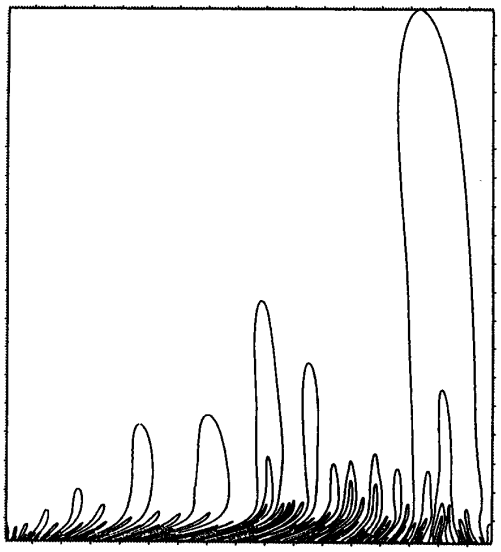

Fig. 15. Renormalized CSS of Africa with nonuniform noise.

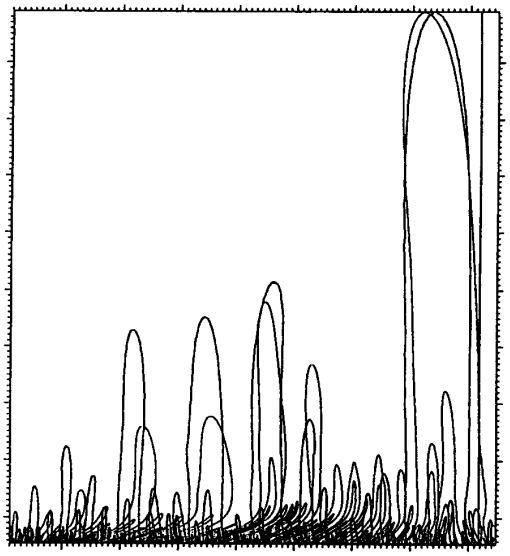

Fig. 16. Superposition of Figs. 14 and 15.

Now, define

$$\hat{X}(w,\sigma) = X(\Phi_\sigma^{-1}(w),\sigma) \qquad \hat{Y}(w,\sigma) = Y(\Phi_\sigma^{-1}(w),\sigma) \tag{3.1}$$

that is, each evolved curve Γ_σ is reparametrized by its normalized arc length parameter w. Notice that

$$\Phi_\sigma(0) = 0 \qquad \Phi_\sigma(1) = 1$$

and

$$\frac{d\Phi_\sigma(u)}{du} = \frac{|\boldsymbol{R}_u(u,\sigma)|}{\int_0^1 |\boldsymbol{R}_v(v,\sigma)|dv} > 0 \text{ at nonsingular points.}$$

In addition, $\Phi_0(u) = u$. $\Phi_\sigma(u)$ deviates from the identity function $\Phi_\sigma(u) = u$ only to the extent to which the scale-related statistics deviate from stationarity along the original curve.

Once we have changed parameters according to (3.1), the curvature of the curve with the normalized path length parameter is given by

$$\kappa(w,\sigma) = \hat{X}_w(w,\sigma)\hat{Y}_{ww}(w,\sigma) - \hat{X}_{ww}(w,\sigma)\hat{Y}_w(w,\sigma).$$

The function defined implicitly by

$$\kappa(w,\sigma) = 0$$

is the *renormalized* curvature scale space image of Γ. Fig. 14 shows the renormalized curvature scale space of Africa, and Fig. 15 shows the renormalized curvature scale space of noisy Africa. Fig. 16 shows the superposition of Figs. 14 and 15. It can be seen that the degree of match of Fig. 14 to Fig. 15 is much better than the degree of match of Fig. 3 to Fig. 13.

IV. The Resampled Curvature Scale Space Image

Note that as a planar curve evolves according to the process defined in Section II, the parametrization of its coordinate functions $x(u)$ and $y(u)$ does not change. In other words, the function mapping values of the parameter u of the original coordinate functions $x(u)$ and $y(u)$ to the values of the parameter u of the smoothed coordinate functions $X(u,\sigma)$ and $Y(u,\sigma)$ is the identity function.

For both theoretical and practical reasons, it is interesting to generalize the definition of evolution so that the mapping function can be different from the identity function. Again, let Γ be defined by

$$\Gamma = \{(x(w), y(w)) | w \in [0,1]\}.$$

The generalized evolution that maps Γ to Γ_σ is now defined by

$$\Gamma \rightarrow \Gamma_\sigma = \{(X(W,\sigma), Y(W,\sigma)) | W \in [0,1]\}$$

where

$$X(W,\sigma) = x(W) \circledast g(W,\sigma) \qquad Y(W,\sigma) = y(W) \circledast g(W,\sigma).$$

Note that $W = W(w,\sigma)$. Furthermore, $W(w,\sigma_0)$, where σ_0 is any value of σ, is a continuous and monotonic function of w. This condition is necessary to ensure physical plausibility since W is the parameter of the evolved curve Γ_σ.

An especially interesting case is when W always remains the arc length parameter as the curve evolves. When this criterion is satisfied, the evolution of Γ is referred to as *arc length evolution*. An implicit equation for W can be derived [9].

Let

$$\boldsymbol{R}(W,\sigma) = (X(W,\sigma), Y(W,\sigma)).$$

The Frenet equations for a planar curve are given by

$$\frac{\partial \boldsymbol{t}}{\partial u} = \left|\frac{\partial \boldsymbol{R}}{\partial u}\right| \kappa \boldsymbol{n} \qquad \frac{\partial \boldsymbol{n}}{\partial u} = -\left|\frac{\partial \boldsymbol{R}}{\partial u}\right| \kappa \boldsymbol{t}.$$

Let $t = \sigma^2/2$. Observe that

$$\frac{\partial}{\partial t}\left(\left|\frac{\partial \boldsymbol{R}}{\partial u}\right|^2\right) = \frac{\partial}{\partial t}\left(\frac{\partial \boldsymbol{R}}{\partial u}\cdot\frac{\partial \boldsymbol{R}}{\partial u}\right) = 2\left(\frac{\partial \boldsymbol{R}}{\partial u}\cdot\frac{\partial^2 \boldsymbol{R}}{\partial u \partial t}\right).$$

Note that

$$\frac{\partial \boldsymbol{R}}{\partial u} = \left|\frac{\partial \boldsymbol{R}}{\partial u}\right| \boldsymbol{t} \qquad \frac{\partial \boldsymbol{R}}{\partial t} = \kappa \boldsymbol{n}$$

since the Gaussian function satisfies the heat equation. It follows that

$$\begin{aligned}\frac{\partial}{\partial t}\left(\left|\frac{\partial \boldsymbol{R}}{\partial u}\right|^2\right) &= 2\left(\left|\frac{\partial \boldsymbol{R}}{\partial u}\right| \boldsymbol{t}\cdot\frac{\partial}{\partial u}(\kappa \boldsymbol{n})\right) \\ &= 2\left(\left|\frac{\partial \boldsymbol{R}}{\partial u}\right| \boldsymbol{t}\cdot\left(\frac{\partial \kappa}{\partial u}\boldsymbol{n} - \left|\frac{\partial \boldsymbol{R}}{\partial u}\right| \kappa^2 \boldsymbol{t}\right)\right) \\ &= -2\left|\frac{\partial \boldsymbol{R}}{\partial u}\right|^2 \kappa^2.\end{aligned}$$

Therefore

$$2\left|\frac{\partial \boldsymbol{R}}{\partial u}\right| \cdot \frac{\partial}{\partial t}\left|\frac{\partial \boldsymbol{R}}{\partial u}\right| = -2\left|\frac{\partial \boldsymbol{R}}{\partial u}\right|^2 \kappa^2$$

or

$$\frac{\partial}{\partial t}\left|\frac{\partial \boldsymbol{R}}{\partial u}\right| = -\left|\frac{\partial \boldsymbol{R}}{\partial u}\right| \kappa^2.$$

Let L denote the length of the curve. Now, observe that

$$\frac{\partial L}{\partial t} = \int_0^L \frac{\partial}{\partial t}\left|\frac{\partial \boldsymbol{R}}{\partial u}\right| du = -\int_0^L \left|\frac{\partial \boldsymbol{R}}{\partial u}\right| \kappa^2 du = -\int_0^1 \kappa^2 dw.$$

Since the value w_0 of the normalized arc length parameter w at a point P measures the length of the curve from the starting point to point P, it follows that

$$\frac{\partial W}{\partial t} = -\int_0^W \kappa^2(U,t) dU$$

and therefore

$$W(w,t) = -\int_0^t \int_0^W \kappa^2(U,T) dU dT + w. \tag{4.1}$$

Note that

$$W(w,0) = w.$$

Note further that for any given value t_0 of t, $W(w,t_0)$ is a monotonically increasing function of w. To see this, observe that after changing the order of integration in (4.1) and applying the chain rule for derivatives, we obtain

$$\frac{\partial W(w.t)}{\partial w} = \left(-\int_0^t \kappa^2(W.T) dT\right)\frac{\partial W(w,t)}{\partial w} + 1$$

or

$$\frac{\partial W}{\partial w}\left(1 + \int_0^t \kappa^2(W.T) dT\right) = 1$$

or

$$\frac{\partial W}{\partial w} = \frac{1}{1 + \int_0^t \kappa^2(W,T) dT}$$

which is always positive for any t. Therefore, $W(w,t)$ is a monotonically increasing function of w for any t.

The function defined implicitly by

$$\kappa(W,\sigma) = 0$$

is the *resampled* curvature scale space of Γ. Since the function $\kappa(W,t)$ in (4.1) is unknown, $W(w,t)$ cannot be computed directly from (4.1). However, the resampled curvature scale space can be computed in a simple way. A Gaussian filter based on a small value of the standard deviation is computed. The curve Γ is parametrized by the normalized arc length parameter and convolved with the filter. The resulting curve is reparametrized by the normalized arc length parameter and convolved again with the same filter. This process is repeated until the curve is convex and no longer has any curvature zero-crossing points. The curvature zero crossings of each curve are marked in the resampled curvature scale space image. Note that the standard deviation of the Gaussian chosen above should be small enough so that the deviation from arc length parametrization after each iteration is negligible. Then, the entire process can be considered to model arc length evolution.

We shall next show that this approximation process converges to the solution of (4.1). Let ϵ be the maximum error in the location of any point of Γ when the arc length evolution of Γ is approximated through the process described above using a Gaussian with standard deviation $\Delta\sigma$. As $\Delta\sigma \to 0$, the change in location of each point of Γ during regular or arc length evolution tends to zero, and therefore, $\epsilon \to 0$. Assume that σ is normalized, and therefore, the largest value of σ used is equal to one. It follows that the process described above performs $1/\Delta\sigma$ iterations until completion and that the total

approximation error δ is given by

$$\delta \approx \frac{\epsilon}{\Delta\sigma}.$$

We need to show that $\delta \to 0$ as $\Delta\sigma \to 0$. Observe that at a point P of Γ

$$\epsilon \approx |(\boldsymbol{r} + \kappa\boldsymbol{n}) - (\boldsymbol{r} + \Delta\boldsymbol{r}_g)| = |\kappa\boldsymbol{n} - \Delta\boldsymbol{r}_g|$$

where κ and $\boldsymbol{n}$ are the curvature and the normal vector at P, $\boldsymbol{r}$ is the position vector of point P, and $\Delta\boldsymbol{r}_g$ is the amount of change in the position vector of P after Gaussian approximation. Let

$$\Delta\boldsymbol{r}_g = \kappa_g\boldsymbol{n}_g$$

where $\boldsymbol{n}_g$ is a unit vector with the same direction as that of $\boldsymbol{r}_g$, and κ_g is equal to the length of $\boldsymbol{r}_g$. $\boldsymbol{n}_g$ and κ_g can be thought of as the normal vector at P and the curvature of an arc of a circle going through P. As $\Delta\sigma \to 0$, the curve segment covered by the Gaussian filter can be approximated by a circular segment with constant curvature. It is easily seen that on an arc of a circle, regular evolution causes the same shrinkage rate at every point and is therefore equivalent to arc length evolution. It follows that as $\Delta\sigma \to 0$, $\kappa_g \to \kappa$, and $\boldsymbol{n}_g \to \boldsymbol{n}$. Therefore

$$\epsilon \approx |\kappa\boldsymbol{n} - \kappa_g\boldsymbol{n}_g|.$$

$\Delta\sigma$ is linearly proportional to the length of the neighborhood covered by the Gaussian filter. We can therefore approximate κ by $\Delta\theta/\Delta\sigma$ and κ_g by $\Delta\phi/\Delta\sigma$, where θ and ϕ are tangent directions at P of Γ and the circular arc specified by $(\boldsymbol{n}_g, \kappa_g)$, respectively. It follows that

$$\delta \approx \frac{\left|\frac{\Delta\theta}{\Delta\sigma}\boldsymbol{n} - \frac{\Delta\phi}{\Delta\sigma}\boldsymbol{n}_g\right|}{\Delta\sigma} = \frac{\left|\frac{\Delta\theta\boldsymbol{n} - \Delta\phi\boldsymbol{n}_g}{\Delta\sigma}\right|}{\Delta\sigma}$$
$$= |\Delta\theta\boldsymbol{n} - \Delta\phi\boldsymbol{n}_g| \to 0 \text{ as } \Delta\sigma \to 0.$$

Therefore, the approximation process described above converges to the solution of (4.1).

Fig. 17 shows the resampled curvature scale space of Africa, and Fig. 18 shows the resampled curvature scale space of Africa with nonuniform noise. Fig. 19 shows the superposition of Figs. 17 and 18. Note that a very close match can be observed when matching Fig. 17 to Fig. 18.

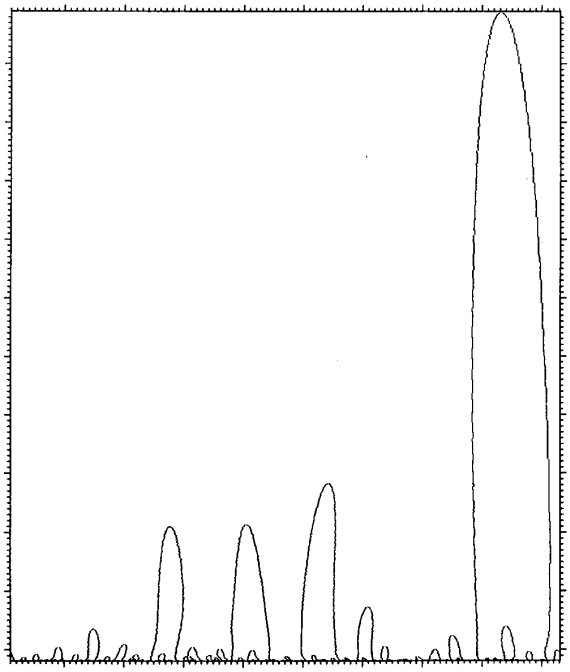

Fig. 17. Resampled curvature scale space image of Africa.

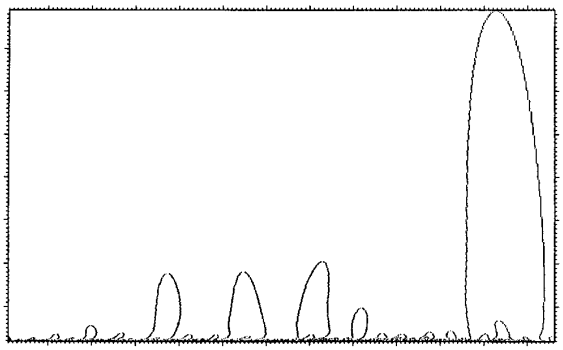

Fig. 18. Resampled curvature scale space image of Africa with nonuniform noise.

V. Evolution and Arc Length

Evolution Properties of Planar Curves

This section presents a number of important results on evolution and arc length evolution of planar curves as defined in Sections II and IV. It also discusses the practical significance of each of those results. The Appendix contains those proofs that have not appeared elsewhere and gives pointers to those that have.

The first five theorems express a number of fundamental properties of evolution and arc length evolution.

Theorem 1: *The order of application of evolution or arc length evolution and a shape preserving transformation (consisting of rotation, uniform scaling and translation) to a planar curve does not change the final result.*

It follows from this theorem that the regular, renormalized, and resampled curvature scale space images of a planar curve satisfy the *invariance* property. The invariance property is essential since it makes it possible to match a planar curve to another of similar shape that has undergone a transformation consisting of arbitrary amounts of rotation, uniform scaling, and translation.

Theorem 2: *A closed planar curve remains closed during its evolution and arc length evolution.*

Theorem 3: *A connected planar curve remains connected during its evolution and arc length evolution.*

Theorems 2 and 3 show that connectedness and closedness of a planar curve are preserved during evolution and arc length evolution. These theorems show that evolution and arc length evolution do not change the physical interpretation of a planar curve as the boundary of a 2-D object. Consider a closed,

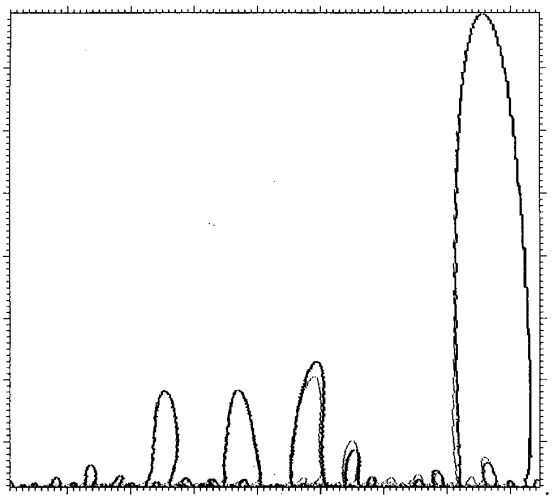

Fig. 19. Superposition of the resampled curvature scale space image of Africa and the resampled curvature scale space image of Africa with nonuniform noise.

connected planar curve that represents the boundary of a 2-D object. If such a curve is not closed and connected after evolution or arc length evolution, then it can no longer be viewed as the boundary of a 2-D object.

Theorem 4: *The center of mass of a planar curve does not move during evolution and arc length evolution of that curve.*

Theorem 5. *Let Γ be a closed planar curve and let G be its convex hull. Γ remains inside G during evolution and arc length evolution.*

Theorem 4 shows that the center of mass of a planar curve remains stationary as the curve evolves, and Theorem 5 shows that a planar curve remains inside its convex hull during evolution and arc length evolution. Together, Theorems 4 and 5 impose constraints on the physical location of a planar curve as it evolves. These constraints become useful whenever the physical location of curves in an image or their locations with respect to each other is important. A possible application area is stereo matching, in which it is advantageous to carry out matching at coarser levels of detail first and then match at fine detail levels to increase accuracy.

Theorem 6 shows that the mapping from a planar curve to its curvature scale space image is an invertible one.

Theorem 6: *Let Γ be a planar curve in C_2. The high-order derivatives at a single point on one curvature zero-crossing contour in the regular, renormalized, or resampled curvature scale space image of Γ determine Γ uniquely up to uniform scaling, rotation, and translation (except on a set of measure zero).*

Theorem 6 shows that the curvature scale space images of a planar curve in fact satisfy the *uniqueness* property. This property ensures that curves of different shapes do not have the same representation. Note that the proof of this theorem is not meant to be used as a practical reconstruction scheme.

Theorem 7 states that under certain conditions, new curvature zero-crossing points are not created during evolution and arc length evolution of planar curves.

Theorem 7: *Let Γ be a planar curve in C_2. If all evolved and arc length evolved curves Γ_σ are in C_2, then all extrema of contours in the regular, renormalized, and resampled curvature scale space images of Γ are maxima.*

Theorem 8 locally characterizes the behavior of planar curves during evolution and arc length evolution just before the creation of a cusp point.

Theorem 8: *Let $\Gamma = (x(u), y(u))$ be a planar curve in C_1, and let $x(u)$ and $y(u)$ be polynomial functions of u. Let Γ_σ be an evolved or arc length evolved version of Γ with a cusp point at u_0. There is a $\delta > 0$ such that $\Gamma_{\sigma-\delta}$ intersects itself in a neighborhood of point u_0.*

The following theorem holds only for arc length evolution.

Theorem 9: *Simple curves remain simple during arc length evolution.*

Theorem 10 locally characterizes the behavior of a planar curve during evolution and arc length evolution just after the creation of a cusp point.

Theorem 10: *Let $\Gamma = (x(u), y(u))$ be a planar curve in C_1, and let $x(u)$ and $y(u)$ be polynomial functions of u. Let Γ_σ be an evolved version of Γ with a cusp point at u_0. There is a $\delta > 0$ such that $\Gamma_{\sigma+\delta}$ has two new curvature zero crossings in a neighborhood of u_0.*

Theorems 8 and 10 together locally characterize the behavior of a planar curve just before and just after the formation of a cusp point during evolution and arc length evolution. This behavior can be used to detect any cusp points that form during evolution or arc length evolution of a planar curve. Such cusp points can then be used effectively to facilitate matching since they provide us with a set of distinctive and easily recognizable features. These theorems also show that self-intersecting curves are described in a natural way by our representation technique. The self-intersection loop gradually grows smaller until it turns into a cusp point and vanishes. In contrast, Asada and Brady's method [1] enlarges the smaller loop until it becomes as large as the larger loop. Figs. 20 and 21 show two self-intersecting curves during evolution. The self-intersection is resolved through the formation of a cusp point, after which the curve becomes simple.

Theorem 7 showed that if a planar curve remains smooth during evolution or arc length evolution, then no new curvature zero crossings will be observed in its curvature scale space images. Theorem 8 showed that every planar curve intersects itself during evolution or arc length evolution just before the formation of a cusp point, and Theorem 9 showed that simple curves remain simple during arc length evolution. Combining Theorems 7–9, we conclude that no new curvature zero-crossing points are created during arc length evolution of simple curves. This is an important result since it indicates that new "structure" is not created in the curvature scale space representations of simple curves [23]. Note that a subclass of self-crossing curves also shares this property.

The result stated by Theorem 9 is also very significant. Simple planar curves usually represent the boundaries of 2-D objects. Arc length evolved versions of those curves can only have physical plausibility if they are also simple. Theorem 9 shows that this is in fact the case. Fig. 22 shows a simple curve

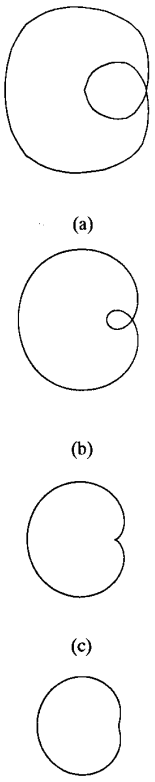

Fig. 20. Convex, self-crossing curve during evolution: (a) Original curve; (b) $\sigma = 20$; (c) $\sigma = 32$; (d) $\sigma = 40$.

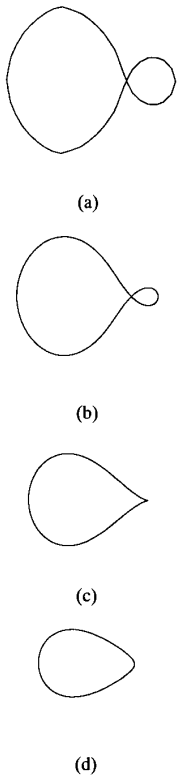

Fig. 21. Concave, self-crossing curve during evolution: (a) Original curve; (b) $\sigma = 16$; (c) $\sigma = 28$; (d) $\sigma = 40$.

and its evolved versions. It can be seen that the curve intersects itself during evolution. Fig. 23 shows the same curve and its arc length evolved versions. As expected, the curve remains simple during arc length evolution.

The final theorem is on the convergence properties of curvature scale space representations.

Theorem 11: *Let Γ be a closed planar curve. Γ becomes simple and convex during evolution or arc length evolution and remains simple and convex.*

Theorem 11 describes a very important property of curvature scale space representations since it shows that the computation of such a representation always has a clearly defined termination point.

VI. Experiments, Discussion, and Evaluation

This section contains additional experiments to demonstrate the stability of the curvature scale space representation. It also contains a discussion of the regular, renormalized, and resampled curvature scale space representations and an evaluation of those representations according to the criteria proposed in Section 1.

Since the representation methods proposed in this paper involve identification of curvature zero-crossing points on planar curves, it may be conjectured that they are not suitable for application to curves with straight segments on them. However, it should be noted that while the presence of straight line segments on a curve might introduce instabilities at the finest scale levels, after a small number of iterations, the originally straight segments will have nonzero curvature, and the computations will stabilize. Fig. 24 shows a planar curve made up of straight line segments, and Fig. 25 shows the curve of Fig. 24 with added random noise. Fig. 26 shows the curvature scale space representation of the curve of Fig. 24, and Fig. 27 shows the curvature scale space representation of the curve of Fig. 25. Fig. 28 shows the superposition of the images shown in Figs. 26 and 27. It can be seen that while there is disagreement between the two representations at the finest scale levels, a very close match exists at the higher levels of the representations.

Three different multiscale representation techniques for planar curves were described in this paper. These three are the regular curvature scale space image, the renormalized curvature scale space image, and the resampled curvature scale space image. Each representation technique is suitable for specific applications. When uniform noise exists on the curve, the regular curvature scale space image can be used. However, when there is nonuniform noise on the curve or when there are local shape differences between the model curves and the image curves, either the renormalized or the

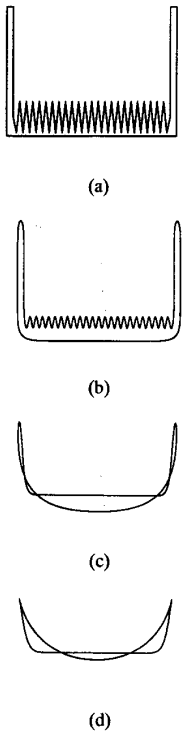

Fig. 22. Simple curve during regular evolution: (a) Original curve; (b) $\sigma = 4$; (c) $\sigma = 16$; (d) $\sigma = 25$.

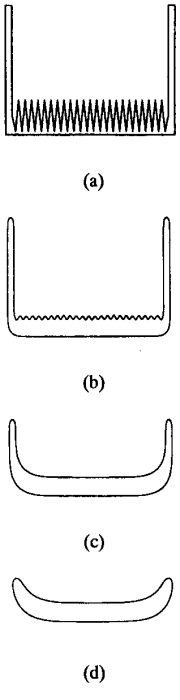

Fig. 23. Curve of Fig. 22 during arc length evolution: (a) Original curve; (b) after six iterations; (c) after 30 iterations; (d) after 50 iterations.

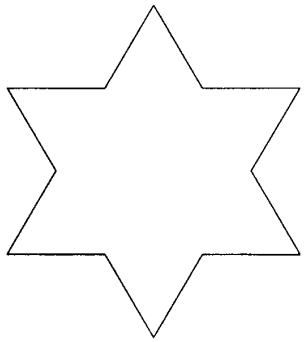

Fig. 24. Planar curve made up of straight line segment.

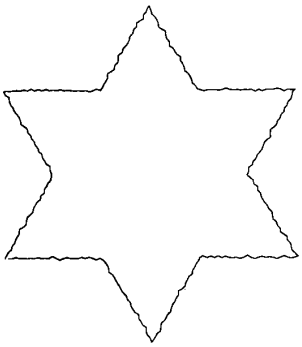

Fig. 25. Curve of Fig. 24 with added random noise.

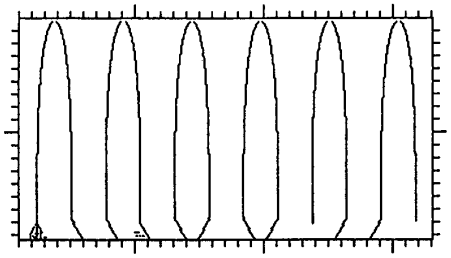

Fig. 26. Curvature scale space image image of the curve of Fig. 24.

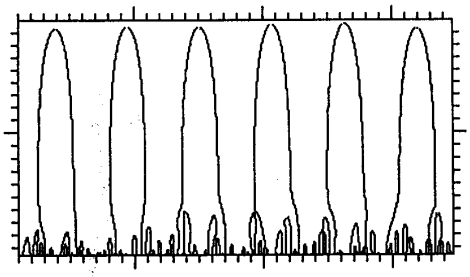

Fig. 27. Curvature scale space image image of the curve of Fig. 25.

resampled curvature scale space images should be used. The renormalized curvature scale space image is the most computationally intensive. Observations indicate that when there are local shape differences, the resampled curvature scale space images show the best overall match, whereas the renormalized

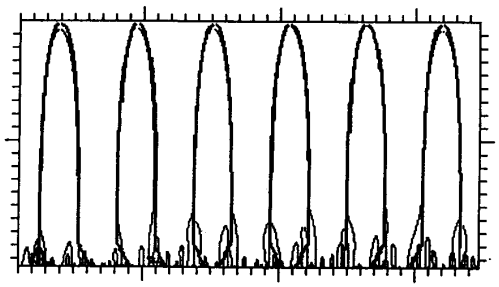

Fig. 28. Superposition of Figs. 26 and 27.

TABLE I
COMPARISON OF REGULAR, RENORMALIZED, AND RESAMPLED CURVATURE SCALE SPACE IMAGES

Representation technique	Advantages	Disadvantages
The Regular Curvature Scale Space Image	• Suitable for transformations consisting of uniform scaling, rotation and translation. • Suitable when uniform, noise corrupts the curve.	• Non-uniform noise or local difference in shape can cause problems.
The Renormalized Curvature Scale Space Image	• More suitable when there is non-uniform noise on the curve or local shape differences exist.	• Most computationally intensive.
The Resampled Curvature Scale Space Image	• Most suitable when there is high-intensity, non-uniform noise or local shape differences exist.	• De-emphasizes shape differences at fine scales.

curvature scale space images match well at high scales but are more influenced by the shape differences at lower scales. Therefore, the choice of the representation technique should depend on the scale level of the curve features that one wishes to emphasize. Table I summarizes the advantages and disadvantages of each representation technique.

We now present an evaluation of the curvature scale space representations according to the criteria proposed in Section I.

Criterion: *Invariance*

Recall that by *invariance*, we meant that the representation for the shape of a curve should not change when shape-preserving transformations (rotation, uniform scaling, and translation) are applied to that curve.

Translation of the curve causes no change in the curvature scale space representations proposed here. Uniform scaling causes the curvature scale space representations to undergo uniform scaling as well. If the represented curves are closed, then their curvature scale space representations can be normalized, and invariance with respect to uniform scaling will also be achieved. If the represented curves are open, changes due to uniform scaling can be handled by a matching algorithm such as the one used in [25].

Rotation causes only a horizontal shift in the curvature scale space representations. However, due to the multiscale nature of those representations, a matching algorithm can determine the shift difference between two matching curvature scale space representations.

Criterion: *Uniqueness*

The *uniqueness* criterion required that two curves with different shapes be mapped to different representations. This is necessary in order to be able to recognize two or more curves with the same shape by observing that their representations are the same.

As argued earlier, Theorem 6 shows that a planar curve can be reconstructed from any of its curvature scale space representations, and therefore, the curvature scale space representations satisfy the uniqueness criterion.

The only arbitrary choice to be made when computing curvature scale space representations is the starting point for parametrization on a closed curve. This only causes a horizontal shift in the curvature scale space representations, but it causes no structural change.

Criterion: *Stability*

The *stability* criterion requires that a small change in the shape of a curve leads to a small change in its representation and vice versa. Theorem 3 shows that planar curves remain connected during evolution and arc length evolution, and therefore, their curvature scale space representations can always be constructed. Furthermore, our observations show that while a planar curve evolves, a small change in the standard deviation of the Gaussian filter always results in a small change in the locations of the curvature zero crossings on that curve. The experiments of this paper also show that curvature scale space representations are stable with respect to significant uniform and nonuniform noise on the curves they represent and therefore satisfy the stability criterion.

Criterion: *Efficiency*

The computation of the representations proposed here calls for evaluation of a large number of convolutions. This process can be rendered efficient using one or more of the following techniques:

1. Fast Fourier transforms
2. parallelism
3. expression of convolutions involving Gaussians of large widths in terms of convolutions involving Gaussians of small widths only
4. tracking curvature zero-crossing points across multiple scales; when it is known that new curvature zero crossings will not be created at higher scales, convolutions can be carried out only in a small neighborhood of the existing zero crossings in order to find the zero crossings at the next higher level.

Furthermore, curvature scale space representations can be stored very efficiently as encoded binary images. An alternative is to store only a selected subset of points from those scale space representations that will be used for matching. In general, all algorithms proposed are efficient in that their complexities are low-order polynomials in the size of the input.

Criterion: *Ease of Implementation*

The procedures needed to compute curvature scale space images are not difficult to implement. Convolutions with Gaussian filters are at the heart of the computations. These are

standard and well-understood procedures in the computational vision area. It follows that this criterion is also satisfied.

Criterion: *Computation of Shape Properties*

The curvature scale space representations of symmetric curves are also symmetric since a symmetric curve also has symmetric curvature zero crossings across scales. Therefore, the symmetry criterion is satisfied. Furthermore, curvature scale space computations are carried out using finite Gaussian filters and making use of finite-sized neighborhoods. Therefore, a curvature scale space representation can also be computed for an open curve and, except near the endpoints of the curve, will resemble the corresponding representation for a closed curve of which it is a part. It is therefore believed that the representations also satisfy the part/whole criterion.

Finally, note that volumetric diffusion [18] and reactive and diffusive deformations of shape [17] are shape representation techniques that are the most similar to ours since they also compute deformations of shapes of planar curves. It may therefore be suggested that an alternative way to compute the curvature scale space representations is to use one of the above techniques to compute deformations of the curves to be represented and then locate curvature zero-crossing points on each deformed curve and map them to the appropriate representations. However, the application of each of the techniques mentioned above might result in disconnected curves. In such cases, it will no longer be possible to construct the curvature scale space representations. Furthermore, our technique combines the curve deformation and the computation of curvature into one step and is therefore more numerically accurate than the aforementioned techniques that separate the processes of curve deformation and computation of curvature.

It follows that the curvature scale space representations satisfy nearly all the criteria considered necessary for a general-purpose shape representation method.

VII. Conclusions

This paper introduced a novel shape representation technique for planar curves and proposed a number of criteria considered necessary for any general-purpose shape representation scheme. Those criteria are *invariance, uniqueness, stability, efficiency, ease of implementation*, and *computation of shape properties*.

Three different ways of computing the representation were described. Each method relies on extracting features of the curve that are invariant under shape-preserving transformations at varying scales. These methods result in the curvature scale space image, the renormalized curvature scale space image, and the resampled curvature scale space image. It was shown that each of those representations is suitable for a specific application.

A number of theoretical properties of those representations were also investigated. These properties together provide a sound foundation for the representations proposed in this paper. Finally, it was shown that the proposed representations satisfy nearly all the criteria introduced earlier.

Appendix

Proof of Theorem 1: The proof of this theorem for regular evolution appeared in [20]. The proof for arc length evolution is very similar. □

Proof of Theorem 2: The proof of this theorem for regular evolution appeared in [20]. The proof for arc length evolution is very similar. □

Proof of Theorem 3: The proof of this theorem for regular evolution appeared in [20]. The proof for arc length evolution is very similar. □

Proof of Theorem 4: The proof of this theorem will be given for arc length evolution only. The proof for regular evolution is very similar.

Let M be the center of mass of $\Gamma = (x(w), y(w))$ with coordinates (x_M, y_M). Then

$$x_M = \int_0^1 x(w)dw \Big/ \int_0^1 dw = \int_0^1 x(w)dw$$
$$y_M = \int_0^1 y(w)dw \Big/ \int_0^1 dw = \int_0^1 y(w)dw.$$

Let $\Gamma_\sigma = (X(W,\sigma), Y(W,\sigma))$ be an arc length evolved version of Γ with $N = (X_N, Y_N)$ as its center of mass. Observe that

$$\begin{aligned} X_N &= \int_0^1 X(W,\sigma)dW \\ &= \int_0^1 \int_{-\infty}^{\infty} g(v,\sigma)x(W-v)dvdW \\ &= \int_{-\infty}^{\infty} g(v,\sigma)\left(\int_0^1 x(W-v)dW\right)dv. \end{aligned}$$

W covers Γ_σ exactly once. Therefore

$$\int_0^1 x(W-v)dW = x_M.$$

Therefore, $X_N = x_M$. Similarly, $Y_N = y_M$. Hence, M and N are the same point. □

Proof of Theorem 5: The proof of this theorem will be given for arc length evolution only. The proof for regular evolution is very similar.

Since G is simple and convex, every line L tangent to G contains that curve in the left (or right) half plane it creates. Since Γ is inside G, Γ is also contained in the same half plane. Now, rotate L and Γ so that L becomes parallel to the y axis. L is now described by the equation $x = c$. Since L does not intersect Γ, it follows that $x(w_0) \geq c$ for every point w_0 on Γ. Let Γ_σ be an arc length evolved version of Γ. Every point of Γ_σ is a weighted average of all the points of Γ. Therefore, $X(W_0,\sigma) \geq c$ for every point W_0 on Γ_σ and Γ_σ is also contained in the same half plane. This result holds for *every* line tangent to G; therefore, Γ_σ is contained inside the intersection of all the left (or right) half planes created by the tangent lines of G. It follows that Γ_σ is also inside G. □

Proof of Theorem 6: The proof of this theorem for the regular curvature scale space image of a planar curve Γ was given in [26].

To prove the same result about the resampled curvature scale space of Γ, recall that derivatives at one point (at any scale) on any curvature zero-crossing contour in the curvature scale space of Γ were computed, and it was shown that the resulting equations can be solved for the coefficients of expansion of the curvature function of Γ in functions related to the Hermite polynomials.

As before, we choose a point on a zero-crossing contour at any scale of the resampled curvature scale space image of Γ and compute the necessary derivatives. The value of σ in the resulting equations is then set to zero. Consequently, the arc length evolved curve Γ_σ, where σ corresponds to the scale at which the derivatives were computed, is reconstructed modulus uniform scaling, rotation, and translation.

The next step is to recover the original curve Γ. This is done by applying *reverse arc length evolution* to Γ_σ. Let the arc length evolved curve Γ_σ be defined by

$$\Gamma_\sigma = \{(X(W,\sigma), Y(W,\sigma)) \mid W \in [0,1]\}.$$

A reverse arc length evolved curve Γ is defined by

$$\Gamma = \{(x(w), y(w)) \mid w \in [0.1]\}$$

where

$$x(w) = X(w.\sigma) \circledast D_N(w.\sigma)$$
$$y(w) = Y(w,\sigma) \circledast D_N(w.\sigma)$$

where D_N is a deblurring operator defined in [15] and

$$w(W,\sigma) = \int_0^\sigma \int_0^w \kappa^2(w,\sigma) dw d\sigma.$$

As a result, Γ is recovered modulo uniform scaling, rotation, and translation.

To prove the same result about the renormalized curvature scale space image, evolved curve Γ_σ is again reconstructed. Then, each of its coordinate functions is deblurred by convolving it with the deblurring operator D_N. Once again, Γ is recovered modulo uniform scaling, rotation, and translation. □

Proof of Theorem 7: Since by assumption all evolved and arc length evolved curves Γ_σ are in C_2, the conditions of the implicit function theorem are satisfied on contours $\kappa(u,\sigma)$, $\kappa(w,\sigma)$, and $\kappa(W,\sigma) = 0$ in the regular, renormalized, and resampled curvature scale space images of Γ. Therefore, the proofs are very similar. A proof for the regular curvature scale space image was presented in [20]. □

Proof of Theorem 8: A proof of this theorem for an arbitrary parametrization of Γ_σ appeared in [20]. It follows that the theorem must also be true of arc length parametrization or close approximations. □

Proof of Theorem 9: Assume by contradiction that Γ is a simple curve that intersects itself during arc length evolution. The location vector of each point of Γ is a continuous function of t during arc length evolution; therefore Γ must touch itself at point P before self-intersection. Let Γ_{t_0} be such a curve. Consider two neighborhoods S_1 and S_2 of Γ_{t_0} that have point P in common. S_1 and S_2 correspond to nonoverlapping ranges of the arc length parameter W. Note that S_1 and S_2 have colinear tangents at P. Let L be the line of those tangents. The tangents exist since it follows from Theorem 8 that P can not be a cusp point on either S_1 or S_2 because Γ_t does not self-intersect for $t \leq t_0$.

Recall that the infinitesimal movement during arc length evolution of each point of S_1 and S_2 is determined by the equation

$$\frac{\partial \boldsymbol{R}}{\partial t} = \kappa \boldsymbol{n}.$$

Therefore, during arc length evolution, every point will move in the direction of the normal vector by an amount equal to the curvature at that point. Similarly, during reverse arc length evolution, every point will move in the opposite direction of the normal vector by an amount equal to the curvature at that point.

It follows that if S_1 and S_2 are on opposite sides of L, after an infinitesimal amount of reverse arc length evolution, they will intersect. This is a contradiction of the assumption that Γ was simple before touching itself. Assume then that S_1 and S_2 are on the same side of L. Note that S_1 and S_2 cannot be overlapping since they would still be overlapping after an infinitesimal amount of reverse arc length evolution, which is also a contradiction of the assumption that Γ was simple before touching itself. Let S_1 be the segment *inside* S_2, i.e., the tangent to S_2 always has S_1 to the same side. It can be seen that S_1 has a larger curvature at P than S_2. Therefore, after an infinitesimal amount of reverse arc length evolution, point P on S_1 and point P on S_2 will move in the same direction, but point P on S_1 will move by a larger amount. It follows that after an infinitesimal amount of reverse arc length evolution, S_1 and S_2 will intersect, which is, again, a contradiction. It follows that Γ remains simple during arc length evolution. □

Proof of Theorem 10: It will be shown that this theorem holds for an arbitrary parametrization of Γ_σ. Therefore, it must also be true of arc length parametrization or close approximations.

Let $(x(u), y(u))$ be an arbitrary parametrization of Γ_σ with a cusp point at u_0. Using a case analysis similar to the one in the proof of Theorem 2 in [20] to characterize all possible kinds of singularities of Γ_σ at u_0, we can again conclude that only the singular points in cases 1 and 4 are cusp points. In case 1, the curve is approximated by (u^m, u^n) in a neighborhood of u_0, where m and n are both even. This type of cusp point cannot arise on Γ_σ if Γ is in C_1. We now turn to the cusp points of case 4. Recall that in case 4, the curve Γ_σ is approximated, in a neighborhood of u_0, by (u^m, u^n), where m is even, and n is odd. Observe that

$$\dot{x}(u) = mu^{m-1} \qquad \ddot{x}(u) = m(m-1)u^{m-2}$$
$$\dot{y}(u) = nu^{n-1} \qquad \ddot{y}(u) = n(n-1)u^{n-2}$$

and

$$\kappa(u) = \frac{\dot{x}(u)\ddot{y}(u) - \dot{y}(u)\ddot{x}(u)}{(\dot{x}(u)^2 + \dot{y}(u)^2)^{3/2}}$$
$$= \frac{mn(n-1)u^{m+n-3} - m(m-1)nu^{m+n-3}}{(m^2u^{2m-2} + n^2u^{2n-2})^{3/2}}$$

since nm, $\kappa(u)$ is always positive on either side of the cusp point in a neighborhood of u_0. Therefore, no curvature zero crossings exist in that neighborhood on Γ_σ.

We now derive analytical expressions for $\Gamma_{\sigma+\delta}$ so that it can be analyzed in a neighborhood of u_0. To blur function $f(u) = u^k$, we convolve a rescaled version of that function with the function $\frac{1}{\sqrt{\pi}}e^{-x^2}$, which is the blurring operator, as follows:

$$F(u) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi}} e^{-x^2} f(u + 2x\sqrt{t})dx$$
$$= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-x^2}(u + 2x\sqrt{t})^k dx$$

where t is the scale factor and controls the amount of blurring. Solving the above integral yields

$$F(u) = \sum_{\substack{p=0 \\ (p \text{ even})}}^{k} 1.3.5\ldots (p-1)\frac{(2t)^{p/2}k(k-1)\ldots(k-p+1)}{p!}u^{k-p}.$$

An expression for $\Gamma_{\sigma+\delta}$ in a neighborhood of the cusp point can be obtained by blurring each of its coordinate functions:

$$X(u) = u^m + c_1 t u^{m-2} + c_2 t^2 u^{m-4} + \ldots + c_{\frac{m-2}{2}} t^{\frac{m-2}{2}} u^2 + c_{\frac{m}{2}} t^{\frac{m}{2}}$$

$$Y(u) = u^n + c_1' t u^{n-2} + c_2' t^2 u^{n-4} + \ldots + c_{\frac{n-1}{2}}' t^{\frac{n-1}{2}} u.$$

Note that all constants are positive, all powers of u in $X(u)$ are even, and all powers of u in $Y(u)$ are odd. It follows that all powers of u in

$$\dot{X}(u) = mu^{m-1} + (m-2)c_1 t u^{m-3} + \ldots + 2c_{\frac{m-2}{2}} t^{\frac{m-2}{2}} u$$

are odd, all powers of u in

$$\ddot{X}(u) = m(m-1)u^{m-2} + (m-2)(m-3)c_1 t u^{m-4} + \ldots + 2c_{\frac{m-2}{2}} t^{\frac{m-2}{2}}$$

are even, all powers of u in

$$\dot{Y}(u) = nu^{n-1} + (n-2)c_1' t u^{n-3} + \ldots + c_{\frac{n-1}{2}}' t^{\frac{n-1}{2}}$$

are even, and all powers of u in

$$\ddot{Y}(u) = n(n-1)u^{n-2} + (n-2)(n-3)c_1' t u^{n-4} + \ldots + c_{\frac{n-3}{2}}' t^{\frac{n-3}{2}}$$

are odd.

The curvature of $\Gamma_{\sigma+\delta}$ in a neighborhood of u_0 is given by

$$\kappa(u) = \frac{\dot{X}(u)\ddot{Y}(u) - \dot{Y}(u)\ddot{X}(u)}{(\dot{X}(u)^2 + \dot{Y}(u)^2)^{3/2}}.$$

Since the denominator of $\kappa(u)$ never goes to zero in a neighborhood of u_0, the zero crossings of $\kappa(u)$ are the same as those of its numerator. Observe that the term with the highest power of u in $\dot{X}(u)\ddot{Y}(u)$ is $mn(n-1)u^{m+n-3}$, and the term with the highest power of u in $\dot{Y}(u)\ddot{X}(u)$ is $m(m-1)nu^{m+n-3}$ and that in both $\dot{X}(u)\ddot{Y}(u)$ and $\dot{Y}(u)\ddot{X}(u)$, all powers of u are even, and all constants are positive. Furthermore, note that at $u = 0$, $\dot{X}(u)\ddot{Y}(u)$ is zero, and $\dot{Y}(u)\ddot{X}(u) > 0$. Therefore, at $u = 0$, $\kappa < 0$. As u grows larger in absolute value, the terms in $\dot{X}(u)\ddot{Y}(u)$ and $\dot{Y}(u)\ddot{X}(u)$ with highest powers of u become dominant (all other terms have positive powers of $t = \delta$ in them). Since the dominant terms have equal powers of u, the one with the larger coefficient becomes the larger term. Since $n > m$, the largest term in $\dot{X}(u)\ddot{Y}(u)$ becomes larger than the largest term in $\dot{Y}(u)\ddot{X}(u)$. Therefore, as u grows in absolute value, κ becomes positive. It follows that there are two curvature zero crossings in the neighborhood of u_0 on $\Gamma_{\sigma+\delta}$. These zero crossings are new since it was shown that no zero crossings exist in the neighborhood of u_0 on Γ_σ. □

Proof of theorem 11: See Mokhtarian [27]. □

ACKNOWLEDGMENT

We are grateful to D. Lowe, J. Little, B. Woodham, D. Kirkpatrick, and the referees for their comments on earlier drafts of this paper that helped improve its quality and presentation.

REFERENCES

[1] H. Asada and M. Brady, "The curvature primal sketch," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, pp. 2–14, 1986.
[2] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Patt. Recogn.*, vol. 13, pp. 111–122, 1981.
[3] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
[4] H. Blum, "Biological shape and visual science (part I)," *J. Theoretical Biol.*, vol. 38, pp. 205–287, 1973.
[5] P. E. Danielsson, "A new shape factor," *CGIP*, vol. 7, pp. 292–299, 1978.
[6] L. S. Davis, "Understanding shape: Angles and sides," *IEEE Trans. Comput.*, vol. C-26, pp. 236–242, 1977.
[7] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Comm. ACM*, vol. 15, pp. 11–15, 1972.
[8] H. Freeman "Computer processing of line-drawing images," *Comput. Surveys*, vol. 6, 1974.
[9] M. Gage and R. S. Hamilton, "The heat equation shrinking convex plane curves," *J. Differential Geometry*, vol. 23, pp. 69–96, 1986.
[10] A. Goetz, *Introduction to Differential Geometry*. Reading, MA: Addison-Wesley, 1970.
[11] R. M. Haralick, A. K. Mackworth, and S. L. Tanimoto, "Computer vision update," Tech. Rep. 89–12, Dept. Comput. Sci., Univ. British Columbia, Vancouver, 1989.
[12] J. Hong and X. Tan, "Recognize the similarity between shapes under affine transformation," in *Proc. ICCV* (Tampa, FL), 1988, pp. 489–493.
[13] B. K. P. Horn and E. J. Weldon, "Filtering closed curves," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, pp. 665–668, 1986.
[14] P. V. C. Hough, "Method and means for recognizing complex patterns," U.S. patent 3 069 654, 1962.
[15] R. A. Hummel, B. Kimia, and S. W. Zucker, "Deblurring Gaussian blur," *CVGIP*, vol. 38, pp. 66–80, 1987.
[16] W. Kecs, *The Convolution Product and Some Applications*. Boston, MA: D. Reidel, 1982.
[17] B. Kimia, A. Tannenbaum, and S. W. Zucker, "Toward a computational theory of shape: An overview," TR-CIM-89–13, McGill Univ., 1989.

[18] J. J. Koenderink and A. J. van Doorn, "Dynamic shape," *Biol. Cybern.*, vol. 53, pp. 383–396, 1986.
[19] D. Lowe, "Organization of smooth image curves at multiple scales," in *Proc. ICCV* (Tampa, FL), 1988, pp. 558–567.
[20] A. K. Mackworth and F. Mokhtarian, "The renormalized curvature scale space and the evolution properties of planar curves," *Proc. IEEE CVPR* (Ann Arbor, MI), 1988, pp. 318–326.
[21] D. Marr, "Representing visual information," AI Memo 415, Mass. Inst. Technol. AI Lab., Cambridge, 1977.
[22] D. Marr and E. C. Hildreth, "Theory of edge detection," *Proc. Royal Soc. London B*, vol. 207, pp. 187–217, 1980.
[23] D. Marr and H. K. Nishihara, "Representation and recognition of the spatial organization of 3D structures," *Proc. Royal Soc. London B*, vol. 200, pp. 269–294, 1978.
[24] J. W. McKee and J. K. Aggarwal, "Computer recognition of partial views of curved objects," *IEEE Trans. Comput.*, vol. C-26, pp. 790–800, 1977.
[25] F. Mokhtarian and A. K. Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, pp. 34–43, 1986.
[26] F. Mokhtarian, "Fingerprint theorems for curvature and torsion zero-crossings," *Proc. IEEE CVPR* (San Diego, CA), 1989, pp. 269–275.
[27] ——, "Convergence properties of curvature and torsion scale space representations," Tech. Rep. 90–14, Dept. Comput. Sci., Univ. British Columbia, Vancouver, 1990.
[28] H. K. Nishihara, "Intensity, visible-surface, and volumetric representations," *Artificial Intell.*, vol. 17, pp. 265–284, 1981.
[29] T. Pavlidis, "Segmentation of pictures and maps through functional approximations," *CGIP*, vol. 1, pp. 360–372, 1972.
[30] ——, "Optimal piecewise polynomial L_2 approximation of functions of one and two variables," *IEEE Trans. Comput.*, vol. C-24, pp. 98–102, 1975.
[31] ——, "Polygonal approximations by Newton's method," *IEEE Trans. Comput.*, vol. C-26, pp. 800–807, 1977.
[32] E. Persoon and K. S. Fu, "Shape discrimination using Fourier descriptors," in *Proc. IJCPR* (Copenhagen, Denmark), 1974, pp. 126–130.
[33] W. Richards, B. Dawson, and D. Whittingham, "Encoding contour shape by curvature extrema," in *Natural Computations* (W. Richards, Ed.). Cambridge, MA: MIT Press, 1988, pp. 83–98.
[34] E. Saund, "Adding scale to the primal sketch," in *Proc. IEEE CVPR* (San Diego, CA), 1989, pp. 70–78.
[35] B. Shahraray and D. J. Anderson, "Optimal estimation of contour properties by cross-validated regularization," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, pp. 600–610, 1989.
[36] J. L. Stansfield, "Conclusions from the commodity expert project," AI Memo 601, Mass. Inst. Technol. AI Lab., Cambridge, MA, 1980.
[37] G. Wahba, "A survey of some smoothing problems and the method of generalized cross-validation for solving them," in *Applications of Statistics* (P. R. Krishnaiah, Ed.). Amsterdam: North-Holland, 1977.
[38] A. P. Witkin, "Scale space filtering," in *Proc. IJCAI* (Karlsruhe, W. Germany), 1983, pp. 1019–1023.
[39] A. L. Yuille and T. A. Poggio, "Scaling theorems for zero-crossings," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, pp. 15–25, 1986.

Farzin Mokhtarian received the B.E.S. degree in Mathematical Sciences from the Johns Hopkins University, Baltimore, MD, in 1982 and the M.Sc. and Ph.D. degrees in computer science from the University of British Columbia, Vancouver, Canada, in 1984 and 1990, respectively.

From 1984 to 1985, he was an Associate Member of the Professional Staff at Schlumberger-Doll Research Laboratory, Ridgefield, CT, where his work focused on 2-D and 3-D seismic image processing. The goal was to locate geological features such as faults, salt domes, and reservoirs to aid in the detection of oil. From 1990 to 1991, he was a Postdoctoral Research Fellow and a Sessional Lecturer at the Department of Computer Science, University of British Columbia. He is currently a Research Scientist at NTT Basic Research Laboratories, Tokyo, Japan. His interests include computational vision, pattern recognition, image processing, computer graphics, computational geometry, and artificial intelligence. His research has focused mainly on 2-D and 3-D shape representation and recognition.

Alan K. Mackworth received the undergraduate degree in Engineering Science from the University of Toronto, Toronto, Canada, in 1966, the A.M. degree in 1967 from Harvard University, Cambridge, MA, and the D.Phil. from Sussex University, Sussex, England, in 1974.

He is currently a Professor of Computer Science at the University of British Columbia and the Shell Canada Fellow of the Artificial Intelligence and Robotics Program of the Canadian Institute for Advanced Research. His research focuses on the theory and applications of computational vision and knowledge representation. Recently, he has worked on developing a framework to determine descriptive and procedural accuracy for visual knowledge representations, 2-D and 3-D shape representations, parallel and distributed constraint satisfaction algorithms, complexity analysis of these algorithms, constraint satisfaction algorithms in schema systems for recognition, visual monitoring of an excavator arm, and model theoretic logical frameworks for depiction, diagnosis, and constraint satisfaction. Applications include diagram understanding, telerobotics, and the interpretation of satelite imagery and geographical sketch maps.

Dr. Mackworth has served as President of the Canadian Society for Computational Studies of Intelligence, as Chair of the Board of Trustees of International Joint Conferences on Artificial Intelligence, Inc., and as founding Coordinator for Computational Perception for the Institute for Robotics and Intelligent Systems. He is the Director of the UBC Laboratory for Computational Intelligence and a Fellow of the American Association for Artificial Intelligence.