

A Unified Curvature Definition for Regular, Polygonal, and Digital Planar Curves

Hairong Liu · Longin Jan Latecki · Wenyu Liu

Received: 6 October 2007 / Accepted: 7 March 2008 / Published online: 29 March 2008
© Springer Science+Business Media, LLC 2008

Abstract In this paper, we propose a new definition of curvature, called visual curvature. It is based on statistics of the extreme points of the height functions computed over all directions. By gradually ignoring relatively small heights, a multi-scale curvature is obtained. The theoretical properties and the experiments presented demonstrate that multi-scale visual curvature is stable, even in the presence of significant noise. To our best knowledge, the proposed definition of visual curvature is the first ever that applies to regular curves as defined in differential geometry as well as to turn angles of polygonal curves. Moreover, it yields stable curvature estimates of curves in digital images even under sever distortions. We also show a relation between multi-scale visual curvature and convexity of simple closed curves.

Keywords Curvature · Multi-scale · Corner detection · Curve evolution

H. Liu (✉) · W. Liu
HuaZhong University of Science and Technology, Luoyu Road,
No. 1037, Wuhan, Hubei 430074, China
e-mail: lhrbss@gmail.com

W. Liu
e-mail: liuwuy@hust.edu.cn

L.J. Latecki
Temple University, Philadelphia, PA 19122, USA
e-mail: latecki@temple.edu

H. Liu
Microsoft Research Asia, Beijing 100080, China

1 Introduction

Curvatures of curves are the key to detect the salient points and to compute the shape descriptors. Mathematically, curvature of a point v is defined as (see Fig. 1):

$$K(v) = \lim_{\Delta S \rightarrow 0} \left| \frac{\Delta\theta(v)}{\Delta S} \right| \quad (1)$$

where $\theta(v)$ is the tangential angle of the point v and S is the arc length.

When applied in digital images, three problems arise:

- (1) The digital images are usually distorted by noise. Fig. 2(a) can be regarded as a pentagram heavily distorted by noise; Fig. 2(b) is the pentagram without noise. For the visual perception, point A is not important, because it should be flat there. However, the curvature computed by formula (1) can be very high.
- (2) The images may have different level of details. If Fig. 2(a) is regarded as an image that looks like a pentagram in global, the curvature of point A should be low in the large scale; at the same time, because there is a very sharp turn in small scale, the curvature should be high.

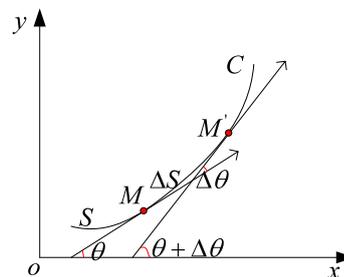


Fig. 1 Curvature of the curve

Obviously, formula (1) is hard to compute the curvature in different scales.

- (3) Due to digitalization, the contours of the images are all stair-like as illustrated in Fig. 3. In such cases formula (1) cannot be directly applied.

We first characterize desirable properties of the curvature that are motivated by human visual perception and are useful in computer vision applications:

- (1) The curvature should be multi-scale and reflect the curviness information of the contour in different scales. In particular, this means that the curvature estimated at a certain level should be able to ignore the influence of small convex and concave parts reflected at lower levels.
- (2) The curvature should be suitable for any planar curves, especially for digital curves; since in computer vision, most of the curves are polygonal curves, in which case formula (1) cannot be directly applied.
- (3) The curvature should be stable under noise.

The main goal of this paper is to propose a new curvature definition that satisfies all above properties. Moreover, the proposed definition applies to both smooth and polygonal curves. To our best knowledge, none of the curvature definitions (including the standard definitions from differential geometry) has this property. Many existing curvature estimation methods are inherently single-scale.

We describe now the main idea of the proposed definition of multi-scale curvature.

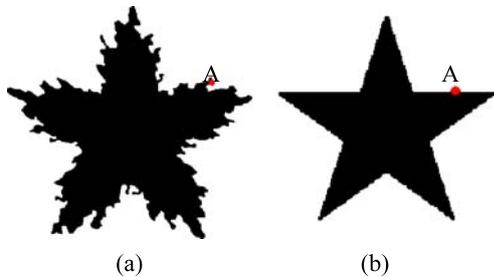
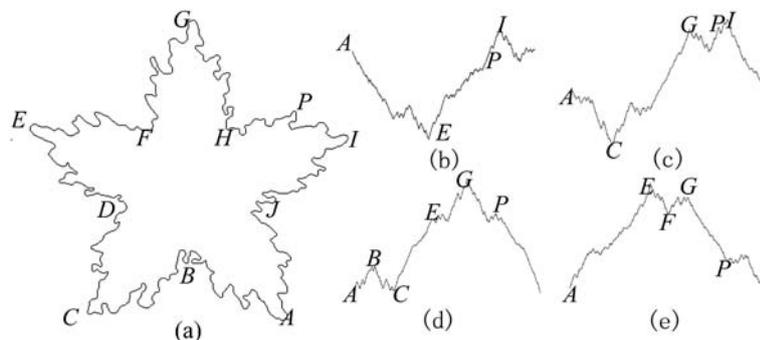


Fig. 2 Pentagram

Fig. 4 The contour of a pentagram and its height functions in 0°, 45°, 90°, 135° directions



We assume that the contour of a 2D shape is parameterized by arc length:

$$C(s) = (x(s), y(s)).$$

We call $x(s)$ the height function in 0° direction and $y(s)$ the height function in 90° direction. Thus, $x(s)$ measures the distance to y -axis in 0° direction. When the coordinate system is rotated by angle α anticlockwise, the new $x(s)$ becomes the height function of the contour in direction α , which we denote H_α .

By rotating the coordinate system by angle $\alpha_i = \pi \frac{i}{N}, i = 0, \dots, N - 1$, we obtain a series of height functions H_{α_i} .

Figure 4(b), (c), (d), (e) shows the height functions of the contour curve in Fig. 4(a) in 0°, 45°, 90°, 135° directions, respectively. Every height function reflects partial information of the contour. The curvature is related to the local extreme points of the height functions: In more directions the point is an extremum, the sharper the contour is at the point, the higher is the curvature of the point. This phenomenon inspires us to estimate the curvature at a contour point v by counting the number of directions in which v is an extremum of the height function.

Obviously, all of the extreme points are not of the same importance. Noise may perturb the curve and cause small extreme points in the height functions. However, a point on a small concave or convex part can not become an extremum of many height functions, while a point on a large concave or convex part will be a clear extremum in many height functions. For example, in Fig. 4(d), point E is not a clear extreme point, but E is a very important minimum point in Fig. 4(b). When the number of height functions is sufficiently large, no important points are ignored, and important high curvature points are detected. In this paper, we

Fig. 3 Stair-like contours



obtain multi-scale curvature by ignoring small heights in the height functions.

The new definition for curvature, called **visual curvature**, is based on statistics of the extreme points of the height functions computed over all directions. Moreover, by gradually ignoring relatively small heights, multi-scale visual curvature is constructed. The multi-scale visual curvature has the following properties:

- (1) It is suitable for every planar curve. On the regular curve, when the number of the height functions approaches infinite, its limit is the standard curvature. On the polygonal curve, it is identical to turn angle.
- (2) It forms a scale space. The scale parameter has a clear geometric meaning: it is a measure of the depth of convex or concave parts.
- (3) The curvature is obtained by ignoring small heights, not by smoothing. Hence it does not change the original curve.

The related literature is reviewed in Sect. 2. In Sect. 3, the visual curvature is defined and its relations to standard curvature and turn angle are proved. In Sect. 4, a scale measure of extreme point is defined in the point of absolute extreme, and its geometric meaning is analyzed. In Sect. 5, some properties of multi-scale visual curvature are described and their significances are discussed. In Sect. 6, some implementation details are analyzed and the experimental results are demonstrated. In Sect. 7, we describe its application in corner detection.

2 Literature Review

Curvature measures how a curve bends, which is one of the most characteristic property of a curve. Cartan (1935) proved that a planar differentiable curve is fully determined by its curvature and the first order derivative of the curvature. Based on his work, Calabi et al. (1998) proposed the concept of “signature curve” as “a new paradigm for the invariant recognition of visual objects”, which is further developed by Boutin (2000).

In computer vision, we usually deal with curves in digital space, i.e., curves extracted from images, which are distorted by noise; however, curvature estimation is known to be very susceptible to noise. Kovalevsky (2001) even pointed that “*under the conditions typical for digital image processing the curvature can rarely be estimated with a precision higher than 50%*”. This is because under the definition of curvature, all these methods have much in common with that of estimating the derivatives of numerical functions. Another problem for traditional definition of curvature with derivatives is that it can not be directly applied to digital curves, as pointed out by Kovalevsky (2001).

Most existing curvature-estimation techniques are under the assumption that there is a unique curvature at each point (Dudek and Tsotsos 1997). In a pure mathematical view, this is obvious true; however, in both human visual perception and in computer vision, the curvature of a point may take on differing values depending on particular goals, e.g., depending whether a given point is regarded as noise or signal point. Some methods can calculate curvatures at multiple scales by two steps: (1) Multi-scale approximation of original curves; (2) Estimate the curvature at approximate curves. In the past few decades, many multi-scale shape representation methods are proposed; we just review some of them which are very relevant to multi-scale curvatures. These methods are highly dependent on multi-scale approximation techniques. One popular way of approximating curves at multi-scale is the convolution with a smooth kernel. Asada and Brady have developed a description (Asada and Brady 1986) they refer to as the “curvature primal sketch” as a fundamental and comprehensive intermediate image representation. The description is a multi-scale structure based on the extraction of changes in curvature. Mokhtarian and Mackworth (1992) proposed a multi-scale, curvature based shape representation technique by convolving the contour with a Gaussian kernel. They proposed a new shape descriptor which is called CSS and demonstrated many of its appealing properties. However, this method modifies the original curve. At the same time, the geometric meaning of its scale factor which is in fact a parameter of Gaussian kernel is not obvious. Based on CSS, Adamek and Connor proposed a method representing contour convexities and concavities at different scale levels (Adamek and Connor 2004). A similar method (Lowe 1988), also using Gaussian kernel, proposed by Lowe, organized the smooth image curves at multiple scales. By instituting Gaussian kernel with dilated-spline kernel, Yu-Ping Wang proposed a new curve smooth scheme (Wang et al. 1999), however, since just kernel function is altered, this method have the same problems as method in Mokhtarian and Mackworth (1992).

Another popular way of shape representation in digital grid is curvature based polygonal approximation (Dudek and Tsotsos 1997; Katzir et al. 1994; Ansari and Delp 1991; Pinheiro et al. 2000; Bengtsson and Eklundh 1991). In these methods, the original contour is approximated by simplified polygon. Obviously, in polygonal arcs, a natural measure of curvature information is turn angle. The problem with standard curvature is that it is defined on smooth curve and can not be applied to polygonal arcs directly. Thus, in all these methods, they need complicated estimation procedure to estimate the curvature of its preimage (which is supposed to be smooth curves with restricted curvature).

Some researchers use spline curves to approximate original curves (Medioni and Yasumoto 1986; Lu and Milios 1991). Based on spline approximation, they calculate the

curvatures and detect the corners (Lu and Milios 1991). These methods are usually single-scale and if the approximation is not good, the precision of estimated curvature is low.

Now we review some other methods for curvature computation. Since a large number of such methods have been proposed, it is beyond the scope of this paper to list all of them. Therefore, we mention only a few beginning with a very influential method from the early days of computer vision (Rosenfeld and Johnston 1973) through methods in Worring and Smeulders (1993), Coeurjolly et al. (2001), Lewiner (2004), Lowe (1988), Yuille (1989), Lewiner et al. (2004), Belyaev (2004), Gumhold (2004), Hermann and Klette (2003). Those approaches can be classified into three groups, according to definition of curvature they are using (as done in Worring and Smeulders 1993): tangent direction, osculating circle, and derivation. Most methods use a sliding window of $2q + 1$ points centered around each point p_j .

Methods based on the tangent direction The first group of methods estimate the derivative of the tangent direction with respect to the arc-length, which is just the definition of curvature. For digital images, this requires to estimate the gradient of a polygonal approximation of an implicit curve. The precision of estimated curvature by methods in this group, such as in Worring and Smeulders (1993) and Gumhold (2004), depends on the precision of estimated tangent direction.

Methods based on Radius of the Osculating Circle The second group of methods compute the curvature by estimating the osculating circle touching the curve. In Calabi et al. (1998), Boutin (2000), Coeurjolly et al. (2001), Hermann and Klette (2003), some representative methods belonging to this group are presented. In Coeurjolly et al. (2001), the radius of the circle passing through p_{j-q} , p_j and p_{j+q} is estimated by:

$$\hat{k}(p_j) = \frac{\angle(p_{j-q}p_j, p_jp_{j+q})}{\|p_{j-q}p_j\| + \|p_jp_{j+q}\|}.$$

This result was improved in Calabi et al. (1998) and Boutin (2000) by the area formula: $\hat{k}(p_j) = \pm \frac{4\Delta}{abc}$, where a , b and c are, respectively, the norm of the vectors p_jp_{j-q} , p_jp_{j+q} , and $p_{j-q}p_{j+q}$.

Method based on Derivative of the curve Methods of this group are based on the first and second derivative estimation of the curve. Basically, they are identical to methods of the first group. The difference is that the methods of the first group begin with tangent line estimation, and then estimate the angles of these lines. The methods in this group are estimating derivatives without geometric constructions of tangent lines. In Worring and Smeulders (1993), the *path*

method obtains the derivatives by a convolution with a derived Gaussian kernel. In Belyaev (2004), the derivatives are estimated as weighted local differences among three points centered at p_j .

Note that all these methods rely on the sliding window, thus they in essence estimate the curvature locally and share the same problem: the size of the sliding window is usually hard to choose; if chosen too small, the curvature is not stable because of noise; if chosen too large, the local geometry of the curve is highly distorted because of smoothing; at both situations, the estimated curvature is not stable and of poor precision.

Hermann and Klette give a comparative study on 2D curvature estimators (Hermann and Klette 2007). According to their experiments, they validate and reinforce the conclusion of Kovalevsky (2001) that in digitized images, the estimated curvature is barely possible with a low error rate, even in high resolution images. Utcke analyzed the error-bounds of curvature (Utcke 2003); one interesting result he pointed out is that, contrary to our intuition, the accurate calculation of the curvature for low-curvature regions is in fact impossible for common image-sizes, while reasonable results under favorable conditions may be obtained for higher-curvature regions.

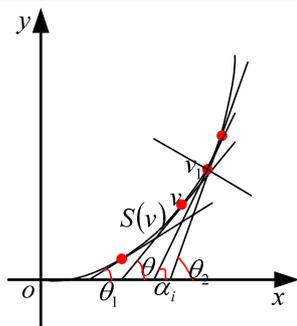
We also mention papers that impose constraints on the digital curves in order to make their shape analysis (in particular, curvature computation) and comparison to their continuous preimages possible. Latecki and Rosenfeld (1998) proposed a class of planar arcs and curves which is general enough to describe (parts of) the boundaries of planar real objects. They analyzed the properties of these arcs and ruled out pathological arcs, thus simplifying the shape representation problem. They also proposed a definition of global curvature that applies to both digital and continuous curves. Gumhold (2004) proposed a curve design system that is based on an optimization algorithm minimizing a variety of optimality criteria.

To summarize, although there are many methods to compute curvature in digital images, they can be viewed as heuristics that apply the standard (continuous) definitions of the curvature to digital curves. Thus, they can not be directly applied to digital or polygonal curves and need to smooth the polygonal arcs (that directly represent digital curves), either by curve fitting or by convolution. This results in parameters that are hard to control, such as the size of sliding window, and displacement of contour points.

3 Visual Curvature

As described in Sect. 1, by rotating the coordinate system, we can obtain a series of height functions H_{α_i} , $\alpha_i = \pi \frac{i}{N}$, $i = 0, \dots, N - 1$.

Fig. 5 The relation between tangential angle and extreme point



Definition 1 For a point v on the curve C , suppose $S(v)$ is its neighborhood of size ΔS on the curve C , the visual curvature of the point v is defined as:

$$K_{N,\Delta S}(v) = \pi \frac{\sum_{i=0}^{N-1} \#[H_{\alpha_i}(S(v))]}{N\Delta S} \tag{2}$$

where $\#[H_{\alpha_i}(S(v))]$ represents the number of local extreme points of the height function H_{α_i} in the neighborhood $S(v)$.

This definition also points out how to compute the visual curvature. For a point v on the contour, we estimate its visual curvature in its small neighborhood $S(v)$. In every height function, we find its extreme points and count the number of the extreme points that are in the neighborhood $S(v)$. We sum up all the numbers and calculate the visual curvature using formula (2). The theorem reveals the relation between visual curvature and the standard curvature on the regular curve. It states that when the number of the height functions is sufficiently large, the visual curvature approaches the standard curvature, denoted by $K(v)$. Regular curve is a curve which is differentiable and the derivative never vanishes.

Theorem 1 For each point v on the regular curve C , we have

$$K(v) = \lim_{\Delta S \rightarrow 0} \lim_{N \rightarrow \infty} K_{N,\Delta S}(v).$$

Proof Let θ be the tangential angle at point v . Assume $\theta \neq 0$. If $\theta = 0$, we rotate the coordinate system so that it satisfies this assumption. Since the curve is regular, by properly rotating coordinate system, there exists a small neighborhood $S(v)$ such that the range of the tangential angle in this neighborhood is a monotonically increasing subset of the half-open interval $[0, \pi)$, devoted by (θ_1, θ_2) , see Fig. 5.

If a point $v_1 \in S(v)$ is the extreme point of the height function H_{α_i} , then $\alpha_i \in (\theta_1, \theta_2)$ and vice versa. Hence the number of the extreme points of all the height functions in the neighborhood $S(v)$ is identical to the number of direction angles α_i that belong to the open interval (θ_1, θ_2) . The direction angle series $\{\alpha_i = \frac{\pi i}{N} | i = 0, \dots, N - 1\}$ of the height functions is a uniform sampling of the half-open

interval $[0, \pi)$. Suppose $\alpha_n = \pi n/N$ and $\alpha_m = \pi m/N$ are the smallest and largest sampling direction angles in the open interval (θ_1, θ_2) , respectively. Then $\pi(m - n + 1)/N$ is an estimation of $\theta_2 - \theta_1$. We now prove that the limit of $\pi(m - n + 1)/N$ is $\theta_2 - \theta_1$ when N approaches infinity. We just need to prove:

- (a) $\pi \lim_{N \rightarrow \infty} \frac{m}{N} = \theta_2$.
- (b) $\pi \lim_{N \rightarrow \infty} \frac{n}{N} = \theta_1$.

Both (a) and (b) can be proved in the same way, thus, we just prove (a). Because $\alpha_m = \pi m/N$ is the largest angle in the set $\{\alpha_i = \frac{\pi i}{N} | i = 0, \dots, N - 1\}$ which is in the interval (θ_1, θ_2) , then $\alpha_m = \pi m/N \leq \theta_2$ and $\alpha_{m+1} = \pi(m + 1)/N > \theta_2$,

$$\lim_{N \rightarrow \infty} |\alpha_m - \theta_2| \leq \lim_{N \rightarrow \infty} |\alpha_m - \alpha_{m+1}| = \pi \lim_{N \rightarrow \infty} \frac{1}{N} = 0,$$

$$\pi \lim_{N \rightarrow \infty} \frac{m}{N} = \lim_{N \rightarrow \infty} \alpha_m = \theta_2.$$

Since (θ_1, θ_2) is a monotonically increasing subset of the tangential angles in $[0, \pi)$, when ΔS is small enough, the value of $\#[H_{\alpha_i}(S(v))]$ is either 0 or 1, thus $\sum_{i=0}^{N-1} \#[H_{\alpha_i}(S(v))] = m - n + 1$.

Therefore,

$$\begin{aligned} \lim_{\Delta S \rightarrow 0} \lim_{N \rightarrow \infty} K_{N,\Delta S}(v) &= \lim_{\Delta S \rightarrow 0} \lim_{N \rightarrow \infty} \pi \frac{\sum_{i=0}^{N-1} \#[H_{\alpha_i}(S(v))]}{N\Delta S} \\ &= \lim_{\Delta S \rightarrow 0} \frac{\pi}{\Delta S} \left(\lim_{N \rightarrow \infty} \frac{m - n + 1}{N} \right) \\ &= \lim_{\Delta S \rightarrow 0} \frac{\theta_2 - \theta_1}{\Delta S} = K(v). \end{aligned}$$

This proves the theorem. □

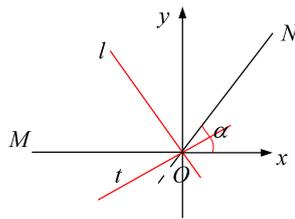
The theorem below reveals the relation between visual curvature and the turn angle of the vertices of polygonal curves. We first motivate this theorem with an example. In Fig. 6, MON is part of the polygonal curve, the turn angle at O is α , t is a line whose directional angle, denoted by β_t , is in the interval $(0, \alpha)$ and l is a line whose directional angle, denoted by β_l , is in the interval (α, π) . Obviously, O is an extreme point of height function in the direction $\pi/2 + \beta_t$ which is perpendicular to t , but it is not an extreme point of height function in the direction $\pi/2 + \beta_l$ which is perpendicular to l . Thus, in Fig. 6, in the direction perpendicular to $\beta \in (0, \alpha)$, O is an extreme point and total range of β is α .

Theorem 2 For a polygonal curve, if O is one of its vertices with turn angle $\alpha(O)$, then

$$\alpha(O) = C \lim_{N \rightarrow \infty} K_{N,C}(O),$$

where C is a constant which is smaller than the minimal distance between two consecutive vertices of the polygonal curve.

Fig. 6 Relation between visual curvature and turn angle



Remark Before we prove this theorem, we make a simple observation. Since $\Delta S = C$ and C is smaller than the minimal distance between two consecutive vertices, we just need to count the number of height functions in which O is an extreme point. This means that we only need to consider $S(O) = \{O\}$ as the neighborhood of O in the case of polygonal curves.

Proof Since C is smaller than the minimal distance between two consecutive vertices of the polygonal curve, we just need to count the number of height functions in which O is an extreme point. Let us assume that there are N height functions and O is an extreme point of M height functions. Then $\pi M/N$ is an estimation of the range of the angle in which direction O is an extreme point. As illustrated in Fig. 6, such range is α . Following the proof of Theorem 1, we obtain: $\sum_{i=0}^{N-1} \#[H_{\alpha_i}(S(O))] = M$, thus

$$\begin{aligned} \alpha(O) &= \pi \lim_{N \rightarrow \infty} \frac{M}{N} = C\pi \lim_{N \rightarrow \infty} \frac{M}{CN} \\ &= C \lim_{N \rightarrow \infty} K_{N,C}(O). \end{aligned}$$

This proves the theorem. □

Theorem 2 reveals that when ΔS is smaller than the minimal distance between two consecutive vertices of a polygonal curve, at vertices of the polygonal curve, visual curvature is proportional to turn angle. Especially, we can assume that $C = 1$, since we can scale the polygonal curve so that $C = 1$. Under this assumption, visual curvature is identical to turn angle.

From Theorems 1 and 2, we know that visual curvature converges to standard curvature or turn angle in different situations, the difference is: for regular curve, $\Delta S \rightarrow 0$ and for polygonal curve, $\Delta S = C$, a small constant. Thus, standard curvature and turn angle are just two special cases of the proposed visual curvature.

The difference between standard curvature and turn angle is obvious: standard curvature relies on two factors, angle and arc length. However, turn angle just relies on angle. Since turn angle does not rely on arc length, it is scale-invariant while standard curvature is scale dependent.

Now for a polygonal curve, at its vertices, we can estimate the standard curvature of underlying regular curve, or we can estimate the turn angle. Especially, when a scale

measure of extreme points is introduced in Sect. 4, we can estimate the multi-scale curvature and multi-scale turn angle, respectively, which are both special cases of multi-scale visual curvature. The difference is: when estimating standard curvature, we must estimate the arc length, which is sensitive to noise; however, when estimating turn angle, we do not need to consider the arc length; thus, in very noisy situation, we can obtain more robust results.

4 A Scale Measure of Extreme Point

In Definition 1, all extreme points are counted, not considering whether they are important or not. Therefore, Theorems 1 and 2 also explain partially why standard curvature and turn angle are not robust. In fact, in a certain scale, small concave or convex parts should be ignored. By imposing a scale measure for extreme point, the multi-scale visual curvature can be defined as follows:

Definition 2 For a point v on a curve C , suppose $S(v)$ is its neighborhood of size ΔS on the curve C , the multi-scale visual curvature of the point v is defined to be:

$$K_{N,\Delta S}^\lambda(v) = \pi \frac{\sum_{i=0}^{N-1} \#[H_{\alpha_i}^\lambda(S(v))]}{N \Delta S} \tag{3}$$

where λ is a scale factor and $\#[H_{\alpha_i}^\lambda(S(v))]$ represents the number of the extreme points of the height function H_{α_i} in the neighborhood $S(v)$ whose scale measure is not smaller than λ (a precise definition follows below, Definition 5). In short, the multi-scale visual curvature is computed by counting the number of relative important extreme points. Note that λ represents scale, the visual curvature at scale λ means that we ignore details with scales less than λ , thus, it is also a scale threshold.

When ΔS is small enough, the value of $\#[H_{\alpha_i}^\lambda(S(v))]$ is either 0 or 1. If point v is an extreme point in direction α_i and corresponding scale measure is not smaller than λ , then the value of $\#[H_{\alpha_i}^\lambda(S(v))]$ is 1, otherwise 0. Formula (3) can be rewritten in the following form:

$$\begin{aligned} K_{N,\Delta S}^\lambda(v) &= \frac{\pi}{N \Delta S} \{ \#[H_{\alpha_0}^\lambda(S(v))] + \#[H_{\alpha_1}^\lambda(S(v))] \\ &\quad + \dots + \#[H_{\alpha_{N-1}}^\lambda(S(v))] \}. \end{aligned}$$

That is, we in fact decompose the visual curvature into N components (N directions), each component is associated with a scale measure. Recall the Fourier Transform in signal theory, if we consider the signals with small scale measure to be high frequency signals, and consider the signals with large scale measure to be low frequency signals, then formula (3) can be regarded as imposing a “low pass” filter on

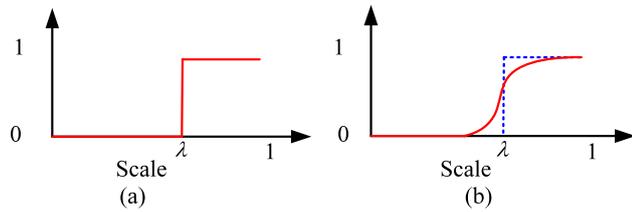


Fig. 7 Two kinds of filters

the visual curvature. Since the value of $\#[H_{\alpha_i}^\lambda(S(v))]$ is either 0 or 1, this “low pass” filter is in fact a stair-like filter, as demonstrated in Fig. 7(a), where λ is the scale threshold.

In signal theory, we usually consider high frequency signals as noises and obtain stable signals by “low pass” filtering. As analyzed above, this is just what we do on the definition of multi-scale visual curvature. Thus, by selecting a proper scale threshold λ , we can get stable visual curvature. There is one problem with stair-like filter: since the value of $\#[H_{\alpha_i}^\lambda(S(v))]$ depends on the relationship between the scale measure and threshold λ , that is, whether the scale measure is larger than or smaller than λ , when the scale measure is close to λ , noise may cause this relationship changes, thus, the value of $\#[H_{\alpha_i}^\lambda(S(v))]$ may suddenly change, from 0 to 1, or from 1 to 0. Just like in signal processing, many other “low pass” filters can be used; these filters change smoothly from 0 to 1, Fig. 7(b) demonstrates one such filter. In this paper, stair-like filter is the default filter.

Just as the importance of frequency in signal theory, the scale measure is very important for multi-scale visual curvature. Note that according to Definition 2, our curvature definition seems to be local; however, our scale measure is defined in global, thus our curvature incorporates the global information. Definition 2 does not impose any constraints on scale measure except that it is a global measure, it just gives a framework; thus, the scale measure can be defined in many different ways. Definition 5 below presents our choice. The intuition is that in every height function, the higher the peak represented by the extreme point is, the more important the extreme point is. We begin with a definition of a measure that quantifies the heights of peaks.

Definition 3 The influence region of a local maximum (minimum) point v in a height function H_α , denoted by $R_\alpha(v)$, is its maximal neighborhood such that the height of every point in this neighborhood is not higher (lower) than the height of the point v . If curve C is open or point v is not an absolute extremum, $R_\alpha(v)$ is divided into two segments by v , we denote the left segment by $R_\alpha^-(v)$ and the right segment by $R_\alpha^+(v)$, see Fig. 8. If curve C is closed, $R_\alpha(v)$ may be the whole curve, in which case $R_\alpha^-(v)$ and $R_\alpha^+(v)$ both represent the whole curve except point v , in particular, $R_\alpha(v) = R_\alpha^+(v) = R_\alpha^-(v)$.

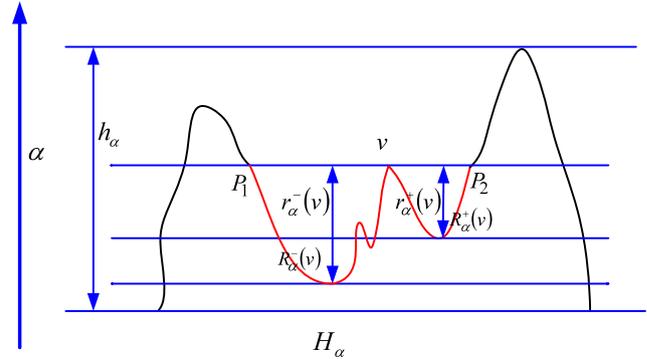


Fig. 8 The influence region and the height of the peak

Definition 4 The height of the peak represented by an extreme point v in the height function H_α , denoted by $r_\alpha(v)$, is defined as:

$$r_\alpha(v) = \min[r_\alpha^+(v), r_\alpha^-(v)],$$

$$r_\alpha^+(v) = \max\{|H_\alpha(p) - H_\alpha(v)| \mid p \in R_\alpha^+(v)\},$$

$$r_\alpha^-(v) = \max\{|H_\alpha(p) - H_\alpha(v)| \mid p \in R_\alpha^-(v)\}.$$

$r_\alpha^+(v)$ and $r_\alpha^-(v)$ are the maximal height differences between v and the points belonging to $R_\alpha^+(v)$ and $R_\alpha^-(v)$, respectively.

In Fig. 8, v is a local maximum point, the curve segment P_1P_2 which is in red is its influence region, the curve segment vP_1 is the left segment of the influence region and curve segment vP_2 is the right segment of the influence region. Obviously, whether a peak is important or not, depends not only on the height of this peak, but also on the scale of the contour or the image. For example, we can scale the whole shape equably, thus scale the value of $r_\alpha(v)$; however, the importance of the peak remains unchanged. There are usually two schemes to normalize $r_\alpha(v)$. One is dividing it by a fixed quantity, no matter what directions; this fixed quantity reflects the scale of the contour or the image, such as the diameter of a shape; we call this scheme “isotropy normalization”. The other scheme is in every direction, dividing it to a quantity which reflects the scale of the contour or the image in this direction; in the definition below, it is divided by the height of H_α , denoted by h_α , which is the height difference between the absolute maximum point and the absolute minimum point of H_α ; we call this scheme “anisotropy normalization”, since in different directions, in this scheme, h_α usually has different values. Of course, there are other normalization methods. In this paper, we choose “anisotropy normalization” scheme as default method, since it results in some good properties which are presented and proved in next section. If we choose other normalization method because of the needs of certain applications, we will explicitly state this.

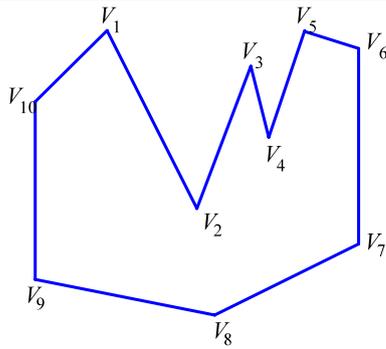


Fig. 9 Example case to discuss the geometric meaning of scale measure

Definition 5 The scale measure of an extreme point v in the height function H_α , denoted by $\lambda_\alpha(v)$, is the height of the peak represented by v divided over the height of H_α :

$$\lambda_\alpha(v) = \frac{r_\alpha(v)}{h_\alpha}. \tag{4}$$

The scale measure of point v represents in which scale in direction α , v can be considered to be important. According to the definition, $\lambda_\alpha(v) > 0$.

If h_α in formula (4) is replaced by the diameter of the shape, which is denoted by D , we get the “isotropy normalization” version definition of scale measure:

$$\lambda_\alpha(v) = \frac{r_\alpha(v)}{D}. \tag{5}$$

Definition 6 The representative scale measure of a point v , denoted by $\lambda(v)$, is the maximum of its scale measures in all the height functions.

For a contour point, according to Definition 2, in the scale larger than its representative scale measure, its visual curvature vanishes and the convex or concave part represented by this point is ignored. Since whether a point is ignored or not at a certain scale depends on its representative scale measure $\lambda(v)$, it is very important to discuss the geometric meanings of $\lambda(v)$.

When isotropic normalization is used (formula (5)), if $r_\alpha(v)$ reaches its maximum, $\lambda_\alpha(v)$ also reaches its maximum. Thus, in this case, we only need to discuss the geometric meaning of $r(v)$, where $r(v) = \max_\alpha \{r_\alpha(v)\}$.

Figure 9 shows a shape with concave parts $V_1 V_2 V_3 V_4 V_5$; it is used to show the geometric meaning of the scale measure. For every point on this shape, we will discuss and demonstrate its geometric meaning of $r(v)$.

In Fig. 9, Points $V_1, V_5, V_6, V_7, V_8, V_9, V_{10}$ are all on the convex hull of this shape, obviously, at these points, the value of $r(v)$ is the maximal distance of the point v to other vertices. For points V_2, V_3, V_4 , the geometric meaning of

$r(v)$ at these points is demonstrated in Fig. 10, it is the distance of these points to corresponding red dash lines. Note that in Fig. 10(a), the red dash line connects point V_1 and V_5 , not point V_1 and V_3 ; this is because point V_2 is the critical point of concave part $V_1 V_2 V_3 V_4 V_5$, not just concave part $V_1 V_2 V_3$. Intuitively, $r(v)$ represents the depth of corresponding convex or concave part.

For anisotropy normalization, the geometric meaning of $\lambda(v)$ is somewhat complicated, since in different directions, $r_\alpha(v)$ is normalized by different values; however, its geometric meaning is similar to isotropy normalization.

5 Properties of Multi-Scale Visual Curvature

This section presents a number of important results on the multi-scale visual curvature. Since multi-scale curvature and multi-scale turn angle are just two special cases of multi-scale visual curvature, there properties are also suitable to them. It also discusses the practical significance of each of those results. The properties below are all under the assumption that the number of the height functions is sufficiently large, thus none of important extreme point is ignored.

Theorem 3 Let C be a closed planar curve and let G be the boundary curve of its convex hull, v is a point on the curve C . Then

- (1) $v \in G$ if and only if $\lambda(v) = 1$.
- (2) $v \notin G$ if and only if $\lambda(v) < 1$.

Proof First, we prove statement (1).

If $v \in G$, there exists a straight line l going through v such that the whole curve C is in one of the half-planes partitioned by l . v is then the absolute extreme point of the height function in the direction perpendicular to l , so $\lambda(v) = 1$.

If $\lambda(v) = 1$, v is then the absolute extreme point of a height function, denoted by H_α . Let l is the straight line going through v perpendicular to α direction, the whole curve C is then in one of the half-planes partitioned by l , so $v \in G$.

Statement (2) is a corollary of statement (1). Since $0 < \lambda(v) \leq 1$, if $v \notin G$, then $\lambda(v) \neq 1$, so $\lambda(v) < 1$. If $\lambda(v) < 1$, then $v \notin G$. This proves the theorem.

In the digital images, the contour C is in fact a polygon with a finite number of vertices $\{V_i | i = 1, \dots, N\}$, where N is the number of the vertices. Let G be the boundary curve of the convex hull of C . For a contour segment defined by vertices $\{V_i | i = m, \dots, n\}$ with V_m and V_n being its two end points, we call it a **maximal concave segment** if all the points except the two end points on the segment do not belong to G .

In Fig. 11, there are two maximal concave segments, $V_1 V_{18} V_{17} V_{16} V_{15} V_{14}$ and $V_8 V_9 V_{10} V_{11} V_{12}$. Obviously, by

Fig. 10 The geometric meaning of $r(v)$ at some contour points

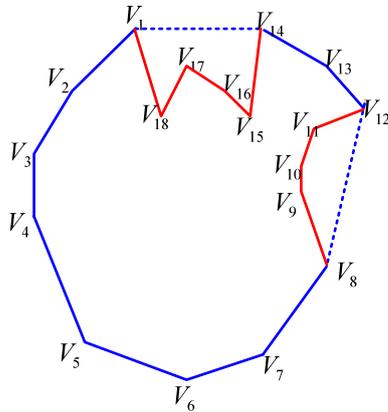
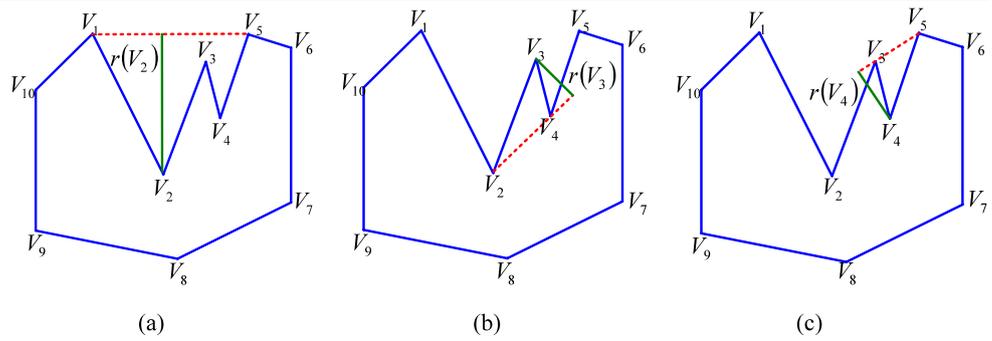


Fig. 11 The concave segments of a polygon

substituting all maximal concave segments with the line segments connecting their two end points, we obtain the convex hull of the polygon. □

Definition 7 The scale measure of a concave segment Γ , denoted by $\lambda(\Gamma)$, is the maximum of the representative scale measures of the points which belong to Γ except the two end points.

In Fig. 11, the scale measures of the two concave segments are:

$$\lambda(V_8 V_9 V_{10} V_{11} V_{12}) = \max\{\lambda(V_9), \lambda(V_{10}), \lambda(V_{11})\},$$

$$\begin{aligned} \lambda(V_1 V_{18} V_{17} V_{16} V_{15} V_{14}) \\ = \max\{\lambda(V_{18}), \lambda(V_{17}), \lambda(V_{16}), \lambda(V_{15})\}. \end{aligned}$$

Since except the two end points, the points which belong to Γ do not belong to the convex hull, according to Theorem 3, $\lambda(\Gamma) < 1$.

Definition 8 Given a scale threshold λ , for a closed polygon C , deletes all the vertices V where the visual curvature vanishes and connects the remaining vertices in sequence. The new polygon is called a λ -scale approximation of C , denoted by C_λ .

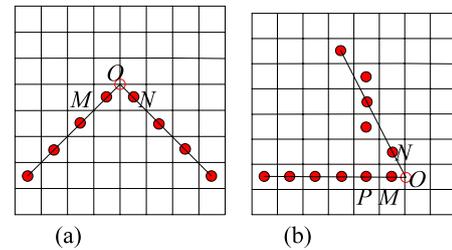


Fig. 12 The pixel configurations in which a high curvature point disappears after digitization

Theorem 4 On the C_λ , all the concave segments which scale measures are smaller than λ are substituted by the line segments connecting their two end points. Specially, $C_0 = C$ and $C_1 = G$.

Theorem 4 is a natural result of Definitions 7 and 8. It shows that as λ increases, more concave segments are ignored and C_λ becomes simpler until it converges to the boundary curve of its convex hull. It also points out how to select the scale threshold λ in the applications: the scale threshold depends on the scale of the concave segments we want to ignore.

6 Implementation Details and Experimental Results

As explained in Sect. 3, different methods of dealing with ΔS result in different visual curvatures. In particular, we can obtain multi-scale curvature and multi-scale turn angle. Since the details of implementation are very similar, thus, we just demonstrate some details of computing multi-scale turn angle. When computing the multi-scale turn angle, as explained before, we let $S(O) = \{O\}$ for each vertex O .

When computing visual curvature, digitalization is the first step. Because of digitalization, some high curvature points may disappear. Let us consider two example cases in Fig. 12.

In Fig. 12(b), the turn angle at point O is about 117° ; however, the original vertex O of the angle formed by two line segments is not represented by a pixel at the same location. The digitalization process mapped it to one of digital

points M and N or possibly to both of them. Neither the turn angle at M nor at N is equal to the turn angle of O , but their sum is. This observation motivates the following approach to compute visual curvature in digital images.

For a given scale λ and a given threshold T , for every point v we consider its neighborhood $U(v)$ of radius T . If the representative scale measure $\lambda(v)$ is the largest among all points in $U(v)$, then the new digital visual curvature at v is sum of all curvature values in $U(v)$, i.e.,

$$DK_{N,\Delta S}^\lambda(v) = \sum_{u \in U(v)} K_{N,\Delta S}^\lambda(u) \quad (6)$$

At the same time we set the digital visual curvature value of all other points in $U(v)$ to zero. Actually, we compute in formula (6) the total visual curvature over the arc determined by the neighborhood $U(v)$, and assign it to a single point. Note that just according to representative scale measure, sometimes we may choose a wrong central point and set the curvature value of correct one to zero, thus, in some applications, such as corner or salient point detection, the true location may be displaced; however, according to our experiments, in the most of cases, the location of corner and salient point is correct.

Now we illustrate on our example in Fig. 12 that our procedure for selecting a digital point whose visual curvature best represents the original point O is correct. After digitalization, O disappears and its information is lost. However, the turn angle of M and N is high in a relative high scale. Point M is selected as best representing O based on the fact that $\lambda(M) > \lambda(N)$ and the distance between M and N is less than T . We show that the proposed approach modifies the turn angle of M to be the turn angle of O in the original continuous contour. For simplicity, let us assume that neighborhood $U(M)$ just contain P, M, N . We obtain

$$DK_{N,\Delta S}^\lambda(M) = K_{N,\Delta S}^\lambda(M) + K_{N,\Delta S}^\lambda(N) + K_{N,\Delta S}^\lambda(P)$$

and $DK_{N,\Delta S}^\lambda(N) = DK_{N,\Delta S}^\lambda(P) = 0$.

The computed digital visual curvature of M is $\pi \times 117/180$. According to Theorem 2, this yields correctly the value of about 117° for the turn angle of M . The justification for this is as follows. In a height function H_α , if one of the points among P, M and N is an extreme point, we increase the count number of M . From the figure, we can observe that if O is an extreme point of the height function of the original continuous object in direction α , then either M or N will be an extreme point of the height function of the digital object in direction α . Thus, the computed turn angle of M is about 117° now. We need to assign 0 to other points in this neighborhood, such as point N and P , since their contribution is added to M . Since parameter T is used to eliminate the negative side-effect of digitalization, and the negative side-effect can be predicted by analyzing possible

similar configurations as demonstrated in Fig. 12, parameter T is usually fixed. In the experiments of this paper, the default value of T is 10. In some cases, we need to choose a smaller T because of the need of the applications; for example, in corner detection, if there are two salient corners very close to each other, we may need to choose a smaller T to detect both of them.

In our method, the main computation load is to compute the scale measures for all the extreme points in all the directions. There are N directions, and in every direction, we just need to be concerned about extreme points. We denote the average number of extreme points by m . Then in the worst case, the time complexity is $O(Nm^2)$. Obviously, for polygonal curves, $m \ll n$, where n is the number of vertices on the curve. Note that even for a regular curve, when it is sampled or digitized, it becomes a polygonal curve. When the scale measures have been computed, given a scale threshold λ , the visual curvature for all the points can be computed in the complexity of $O(n)$.

As pointed out in Sect. 3, for a polygonal curve, at its vertices, we can compute the multi-scale curvature of underlying regular curve, or we can compute the multi-scale turn angle. There are all multi-scale visual curvatures. The only difference is whether we compute ΔS for each vertex. Since ΔS is local and very sensitive to noise, thus, in very noisy situation, we usually use multi-scale turn angle. In our experiments below, we first demonstrate the multi-scale curvature for the regular curve, and then we demonstrate the power of multi-scale turn angle in very noisy situations.

Figure 13 shows the multi-scale curvature of a regular curve, whose polar equation is: $r = 1 - 0.8 \cos(4\theta)$. We sample this curve by evenly sampling θ in the interval $[0, 2\pi)$, the number of sampling points is 628. We first compute the curvature at each sample point by derivatives as ground truth. To compute standard curvature by our method, two factors must be considered: N and ΔS . In this experiment, we use a simple method to estimate ΔS : for a vertex of the polygonal curve sampled from the regular curve, suppose the lengths of the two edges adjacent to a vertex are a and b , respectively, we simply let $\Delta S = (a + b)/2$; Fig. 13(b) shows that at scale $\lambda = 0$, as N increases, the error between computed visual curvature and the curvature computed by derivatives decreases; however, no matter how large N is, we cannot reduce the error to zero, since some errors result from the error in estimating ΔS . From Fig. 13(b), we also find that as N increases, the influence of N quickly decreases; which implies that we do not need very large N . Figures 13(c) and (d) show the computed curvature as arc length function at scale 0.03 and 0.4, respectively. Corresponding multi-scale approximations of this curve are demonstrated in Figs. 13(e) and (f), respectively. The four larger peaks in Figs. 13(c) and (d) correspond to concave points while the four smaller peaks correspond to convex

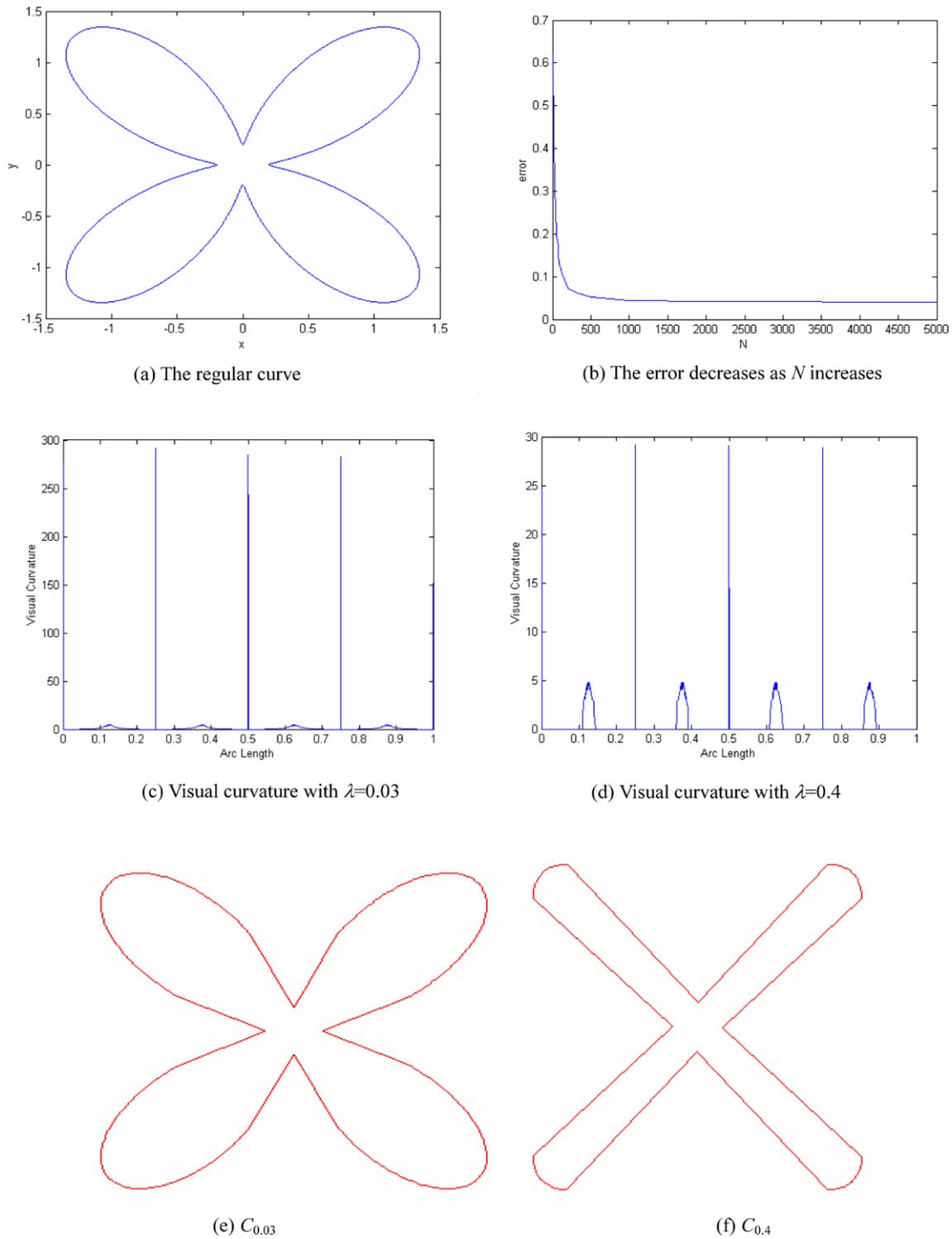


Fig. 13 Multi-scale visual curvature of a regular curve

points. By looking at plots in Figs. 13(c) and (d), one can observe that as λ increases, the curvature value decreases.

Since N only affects the precision of computed visual curvature, in all our experiments below, we set $N = 128$. As

demonstrated in Fig. 13, when estimating standard curvature, part of error results from the error of estimating ΔS , which is very sensitive to noise, thus, in noisy situation, we put our focus on multi-scale turn angle; because of this rea-

Fig. 14 Visual curvatures in different scales

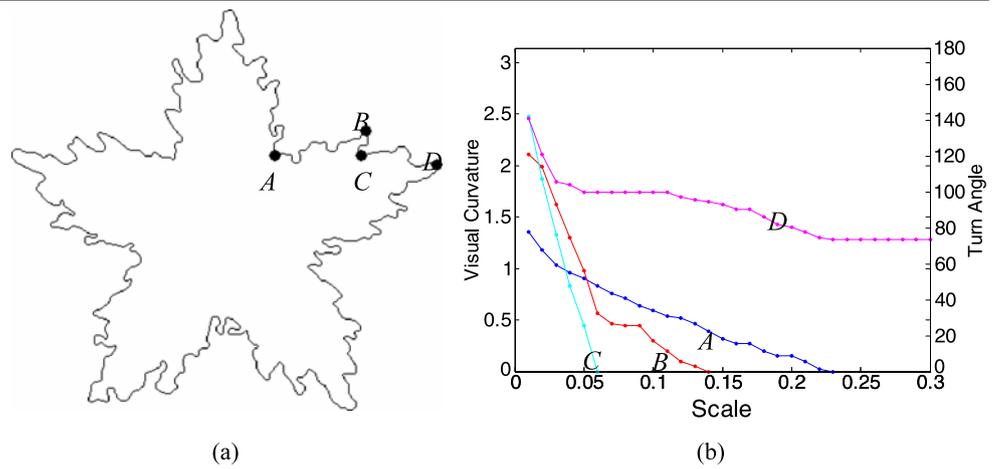
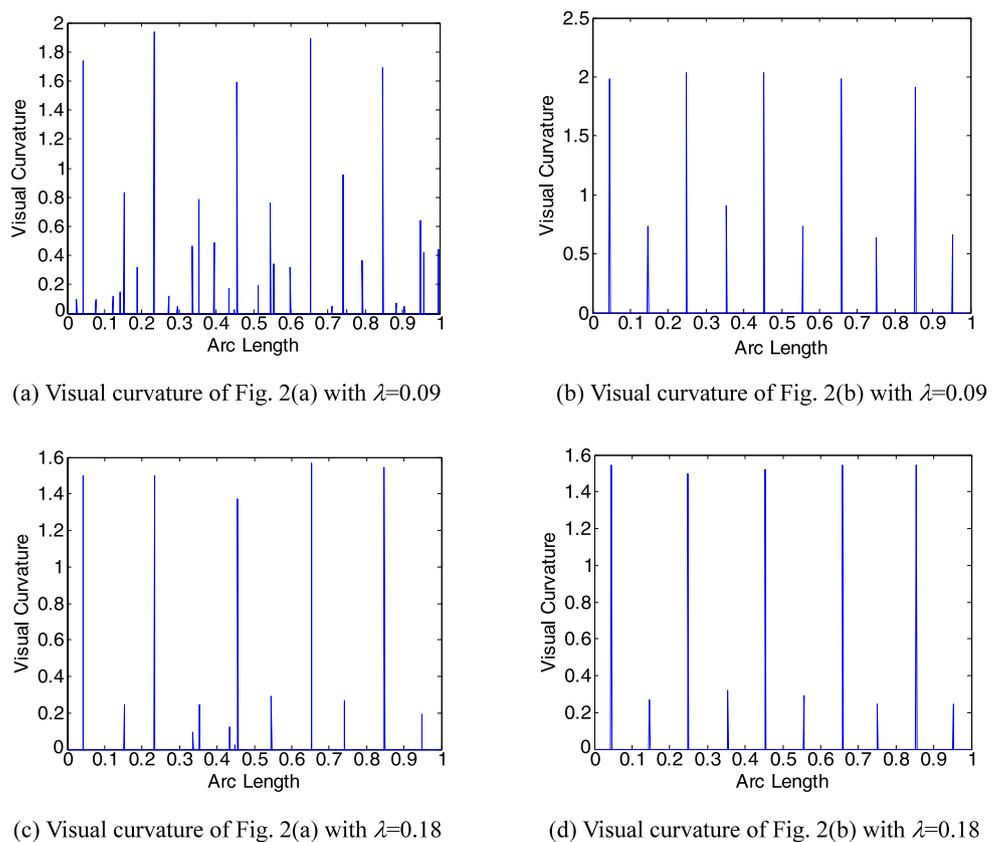


Fig. 15 Visual curvature for the two pentagrams in Fig. 2



son, in the following experiments, visual curvature refers to turn angle.

Figure 14(b) shows the multi-scale turn angle of the four points *A*, *B*, *C*, *D* in the Fig. 14(a) calculated in different scales. Obviously, the turn angle of these four points on a pentagram without noise should be 72° , 0° , 0° and 144° , respectively. Because of noise, *B* and *C* have a large turn angle in small scales. For example, when $\lambda = 0.01$, the turn angle of these points are 77° , 121° , 142° , 140° , respectively. As scale increases, turn angle decreases. Since $\lambda(C) < \lambda(B) < \lambda(A) < \lambda(D)$, the turn angle of *C* vanishes

first, then *B* and *A*, the turn angle of *D* never vanishes since it's a point on the convex hull of the pentagram. As the value of λ increases, the obtained turn angle estimation is not accurate. However, increasing λ is very useful for dominant point detection, e.g., as can be seen in Fig. 14(b). The most dominant point is *D* and then *A*.

Figure 15 demonstrates the multi-scale turn angle as arc length functions for two pentagrams in Fig. 2 in two scales. *A* is a start point and we follow the contour clockwise. Obviously, there are ten peaks in all graphs; the noise in

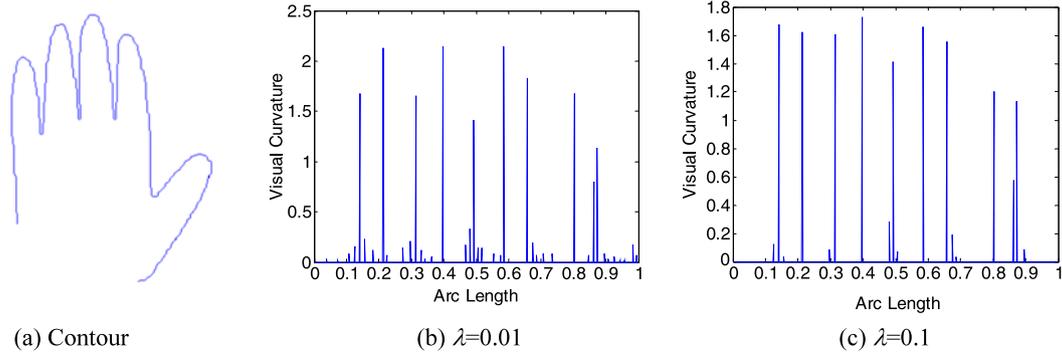


Fig. 16 Part of the contour of a hand and its visual curvatures

Fig. 17 Multi-scale approximation of the shape in Fig. 4(a)

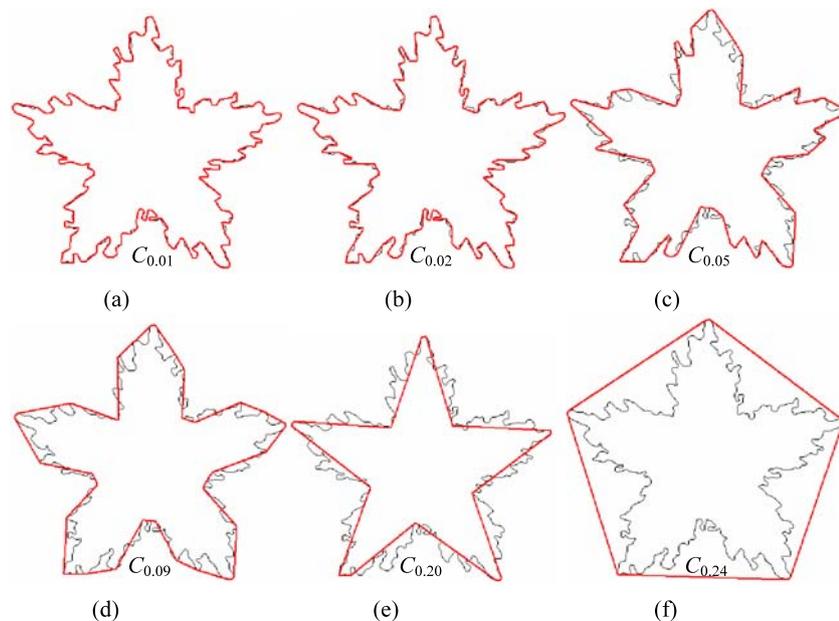


Fig. 2(a) is suppressed, especially when the scale is large, see Figs. 15(a) and (c).

Our method can also be applied to open curves without modifications, only at two end points, the curvature is undefined. Figure 16 demonstrates part of the contour of a hand and its visual curvature as arc length function at scale 0.01 and 0.1.

Figure 17 demonstrates the multi-scale approximation of the shape in Fig. 4(a) obtained by connecting the points with representative scale measure larger than λ . As λ increases, the visual curvature of more points vanishes and C_λ becomes simpler until it converges to the boundary curve of its convex hull.

7 Application in Corner Detection

Corner points, which have high curvature on the contour, play very important roles in shape analysis. It is believed

that most of information on a contour is concentrated at its corner points.

There are usually five criteria to evaluate corner detectors, for a detailed description about corner detector, please refer to Marji (2003):

1. All “true corners” should be detected and no “false corners” should be detected.
2. Corner points should be well localized.
3. Detector should have a high repeatability rate (good stability).
4. Detector should be robust with respect to noise.
5. Detector should be computationally efficient.

Corner points have high curvature, but not all high curvature points are corner points. Noise usually results in high curvature points in small scale; at the same time, high curvature points in small scales are usually not as important as the high curvature points in large scale. Therefore, in our

Fig. 18 The evolution procedure of a horse

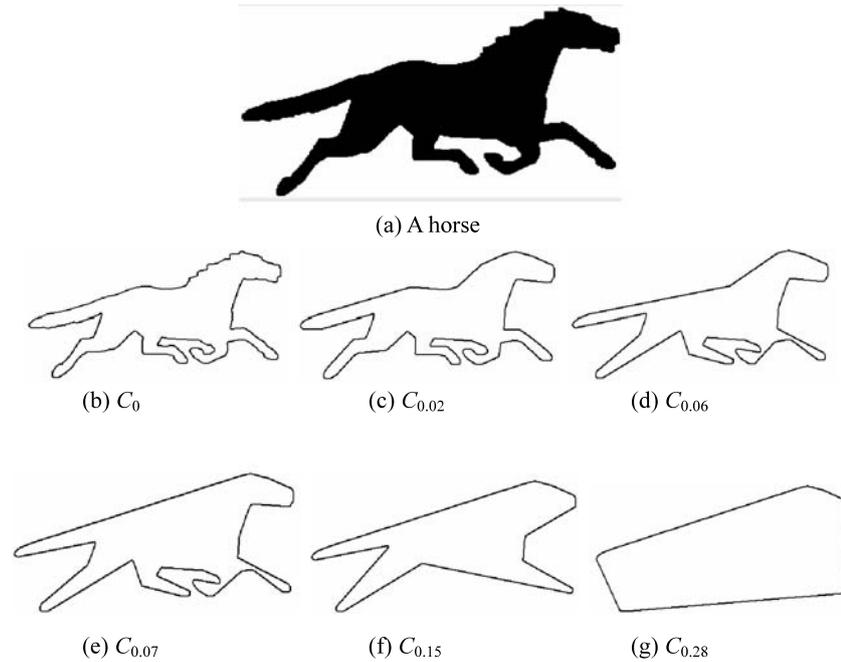
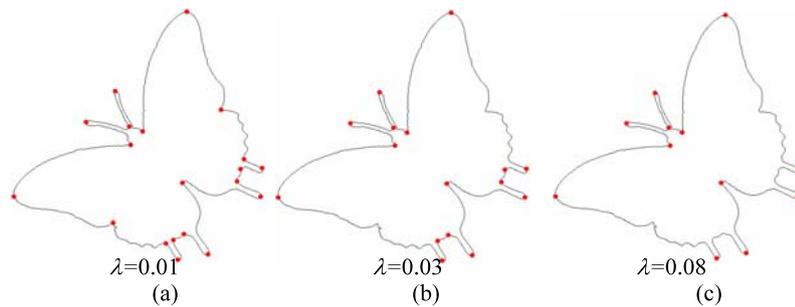


Fig. 19 The corners of a butterfly detected in different scales



framework, we define corners as high curvature points at a relative large scale.

In the past few decades, many corner detectors have been reported. In general, there are two approaches to this problem. One is to detect the corner points directly through angle or corner detection schemes (Rosenfeld and Johnston 1973; Rosenfeld and Weszka 1975; Freeman and Davis 1977; Sankar and Sharma 1978; Anderson and Bezdek 1984; Cederberg 1978; Kruse and Rao 1978). The other approach is to obtain a piecewise linear polygonal approximation of the digital curve subject to certain constraints on the goodness of fit (Pavlidis 1980; Dunham 1986); corner points then correspond approximately to the actual or extrapolated intersections of adjacent line segments of the polygon. Many methods are local; they estimate the angle or approximate the curve in the neighborhood of the central point. Hermann and Klette attempt to detect the corner based on global curvature estimation (Hermann and Klette 2005) and reported better results.

Many existing corner detectors are single scale, thus, they just work well for objects that have similar size features.

However, most of objects consist of multiple size features. Hence they may either miss “true corners” or detect “false corners”. To overcome this drawback, multi-scale scheme is needed. Rattarangsi proposed a multi-scale corner detector based on Gaussian scale space (Rattarangsi and Roland 1992) and demonstrated very impressive results. Mokhtarian and Suomela proposed a multi-scale corner detector based on CSS (Mokhtarian and Suomela 1998), they first extract the contours from grey images by Canny edge detector, then find the high curvature points on large scale. Since this method modifies original curves, they need to track the point location through several lower scales to improve localization.

Our method can estimate the visual curvatures at a continuum of scales. By selecting a relative large scale, the noise can be suppressed and thus “false corners” are eliminated. Since our method does not modify the original curve, the detected corner points are well localized, and there is no need for tracking the corner points over the scales. Note that in the application of corner detection, when we refer to visual curvature, we always mean the multi-scale turn angle.

Fig. 20 Eight shapes used in the tests

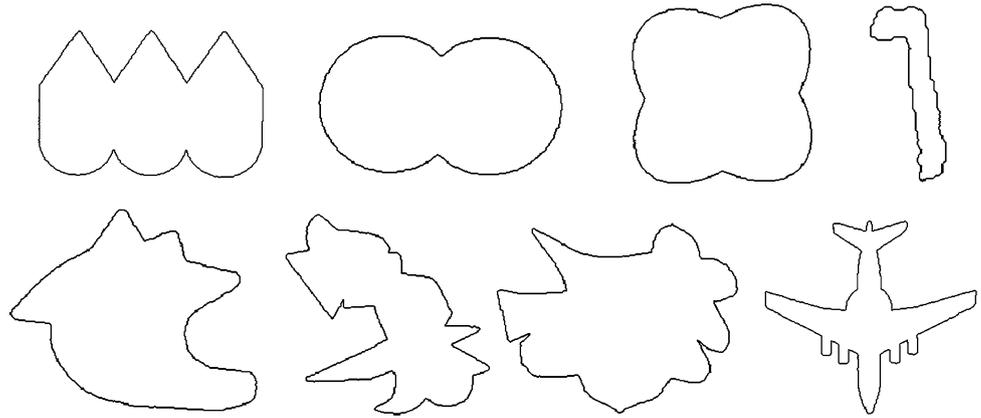
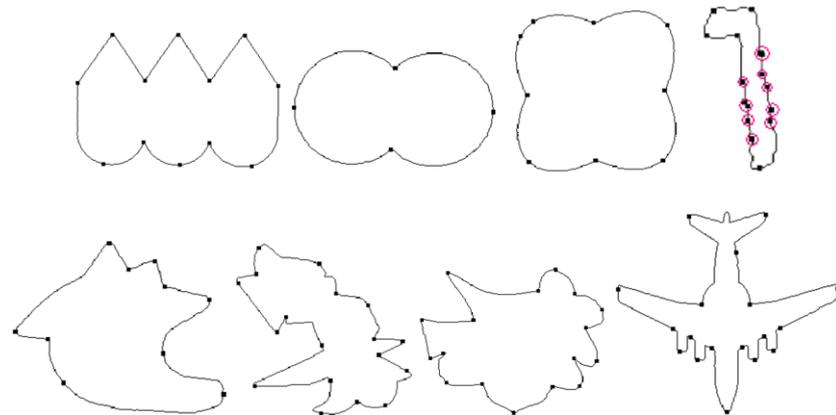


Fig. 21 Results of RJ73



Cong and Parvin (1998) pointed out that certain kinds of curve evolution can be used to enhance the corners. As the curve evolves, small convex or concave parts disappear, similar to skeleton pruning method (Bai et al. 2007). In Definition 8, we define the multi-scale approximation of contour using multi-scale visual curvature, which is in fact a curve evolution method (without displacing the curve points).

Figure 18 shows the evolution procedure of a horse by gradually deleting the points where visual curvature vanishes. As λ increases, some less important corners disappear, but important corners remain, which illustrates the fact that our method can distinguish “true corners” from “false corners”.

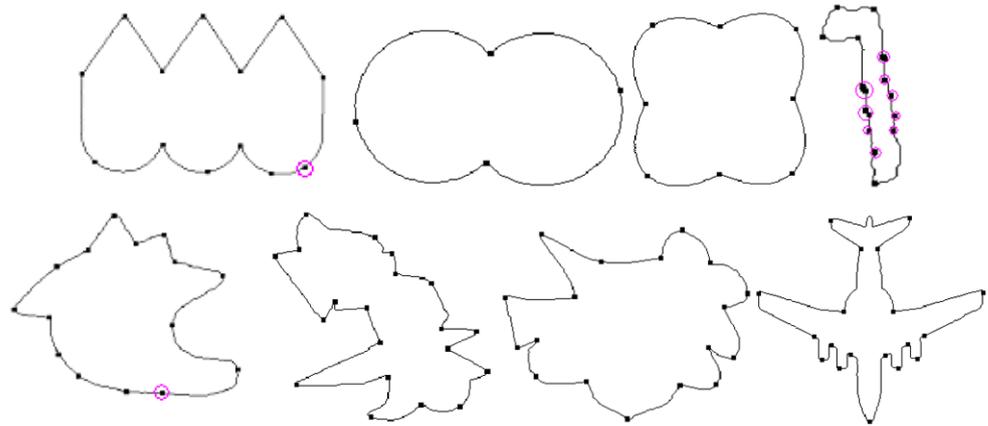
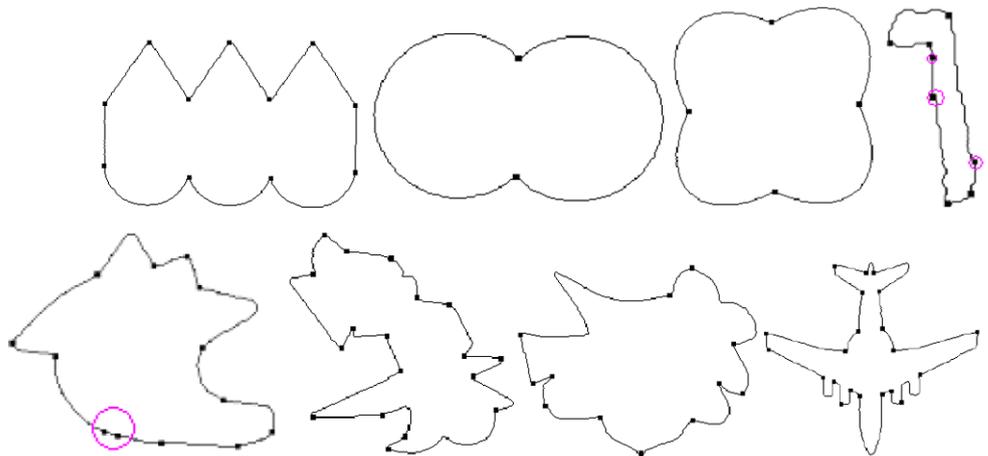
It is a hard task to compare different corner detectors, since whether a point is a “true corner” or a “false corner” sometimes depends on the applications. Thus, after multi-scale visual curvature of a curve is calculated, we have different schemes to choose corner points. In our method, a contour point is described both by its visual curvature and corresponding scales. The simplest scheme is: in a relative large scale λ , we consider the points whose digital visual curvature is above a threshold DK_0 as corner points. Note that in many other corner detectors, they regard the curvature maxima as corner point and thus not need the curvature threshold parameter; however, our experiments show that

there are points which are curvature maxima, but have low curvature value at relative large scales. In our scheme, parameter λ is easy to set; it is used to eliminate the influence of noise and remove small details, thus, it only depends on the strength of noise or small details we want to eliminate. As for how to select parameter DK_0 , according to the prior information we know, there are at least three methods: first, we can set DK_0 directly, according to how large the curvature the point is considered to be corner points; second, we can limit the number of corner points and then set DK_0 accordingly; third, as DK_0 increases, the number of corner points decreases, we can select the curvature where the number of corner points varies slowly as DK_0 .

Figure 19 demonstrates the corners of a butterfly detected in different scales. In this experiment, curvature threshold $DK_0 = 17\pi/64(48^\circ)$.

Obviously, the lower scale we choose, the more corner points are detected, the number of corner points in Fig. 19(a), (b), (c) are 18, 16, 12, respectively. Another important fact is that the corners detected in higher scales remain in the corner set of lower scales.

We compare our method with five other corner detectors on eight test images which are from Chetverikov and Szabo (1999), the other five corner detectors are Rosenfeld and Johnston RJ73 (Rosenfeld and Johnston 1973), Rosen-

Fig. 22 Results of RW75**Fig. 23** Results of FD77

feld and Weszka RW75 (Rosenfeld and Weszka 1975), Freeman and Davis FD77 (Freeman and Davis 1977), Beus and Tiu BT87 (Beus and Tiu 1987) and IPAN99 (Chetverikov and Szabo 1999). We did not implement them, we used the online version of these algorithms <http://visual.ipan.sztaki.hu/corner/cornerweb.html> provided by Chetverikov. In summary, these algorithms all have two procedures: “Corner Strength” estimation and “Corner Selection”. Note that “Corner Strength” is closely relative to curvature. These five detectors all have parameters which are proportional to arc length, these parameters decide the minimum spacing of adjacent corner points; FD77, BT87 and IPAN99 have another parameter similar to “curvature threshold” in our method. The eight test shapes are shown in Fig. 20. Figures 21, 22, 23, 24, and 25 show the results of these five corner detectors. We are aware of the fact that the existence or nonexistence of certain corners is very subjective, but a consistent decision can be made in clear cases. We marked obviously wrong corners in Figs. 21–25 with circles. In addition, these methods miss many important corners, e.g., on the sharp peak at the top left of the seventh shape in Figs. 23 and 24 and on the tail of the plane (the eighth shape) in Figs. 21–24.

Figure 26 shows the detected corners using our method. In this experiment, $\lambda = 0.01$, since the noise is small. The curvature threshold DK_0 is $21\pi/128(30^\circ)$; only for shape 5, DK_0 is $28\pi/128(39^\circ)$. For the fourth shape, our corner detector finds the six true corners, no false corner is found, because at scale 0.01, small noise is suppressed; however, the other five detectors all detect some false corners, these false corners are true curvature maxima, but they are not important shape features; In Figs. 21–25, for the fourth shape, these false corners are marked by red circles.

Figure 27 shows the detected corners using our method at two scales 0.07 and 0.11 with curvature threshold DK_0 is $21\pi/128(30^\circ)$. As we stated, the scale is in fact a measure of importance of corner points. In Fig. 26, at scale 0.01, all corners around the four engines of the plane are detected; In Fig. 27(a), at scale 0.07, for each engine, we just detect two corners, one at the convex part and the other at the concave part; at even large scale 0.11, shown in Fig. 27(b), there are no corners for engines, since engines are considered to be small details at this scale. Obviously, this makes sense for shape description.

In our method, the geometric meaning of the representative scale measure is the depth of corresponding convex or

Fig. 24 Results of BT87

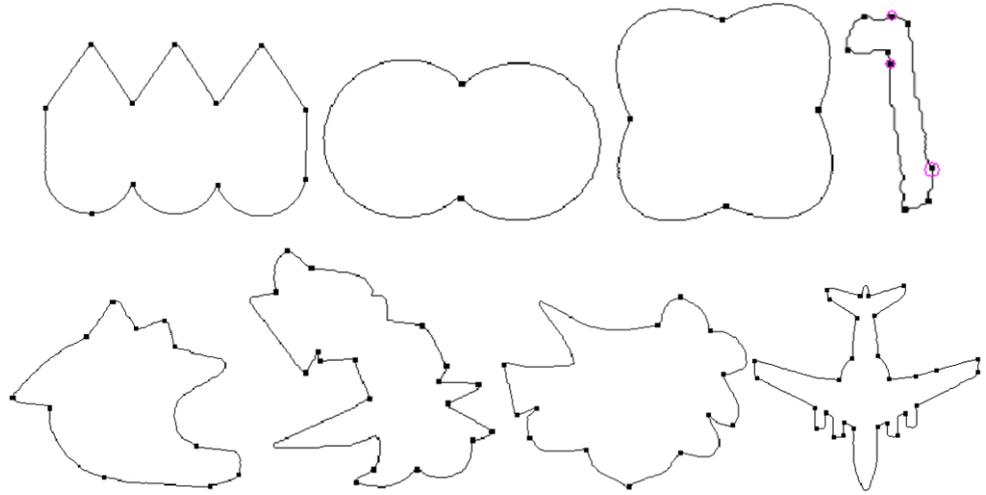


Fig. 25 Results of IPAN99

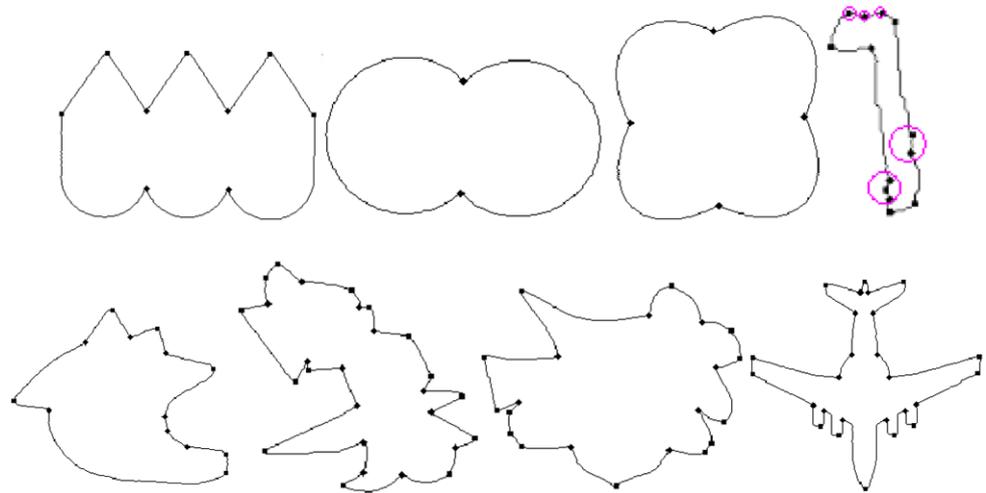


Fig. 26 Results of our method

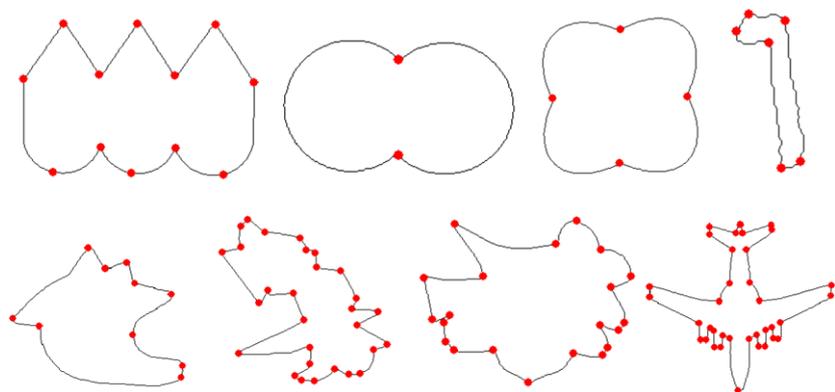


Table 1 The scale parameters for the experiments in Fig. 28

No.	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)
Parameters	0.04	0.03	0.07	0.03	0.07	5	15	(4,7)	(10,13)	8	30

concave part; thus, we can suppress relative large noise or shape details by calculating the curvature at a larger scale.

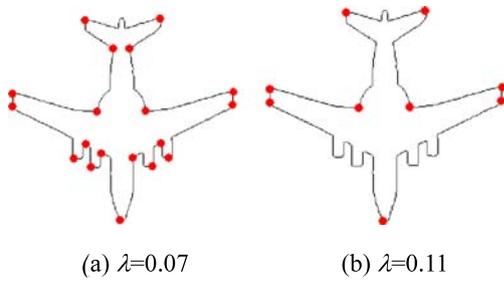


Fig. 27 The corners of the plane test images at different scales

This is the main advantage of our method in the presence of noise. The five other corner detectors all have parameters which are proportional to arc length, these parameters decide the minimum spacing of adjacent corner points. According to our experiments, by increasing these parameters some large noise can be suppressed, thus, these parameters function like scale parameters; however, they are definitely not good scale parameters, since there may be important corners which are very close to each other. To illustrate the detected corners in such a situation, we add significant noise to the eighth test images (plane) and the results are shown in Fig. 28.

Since the noise is significant, in our method, $\lambda = 0.04$, the result is shown in Fig. 28(a). For other five corner detec-

Fig. 28 The detected corners for plane image with significant noises

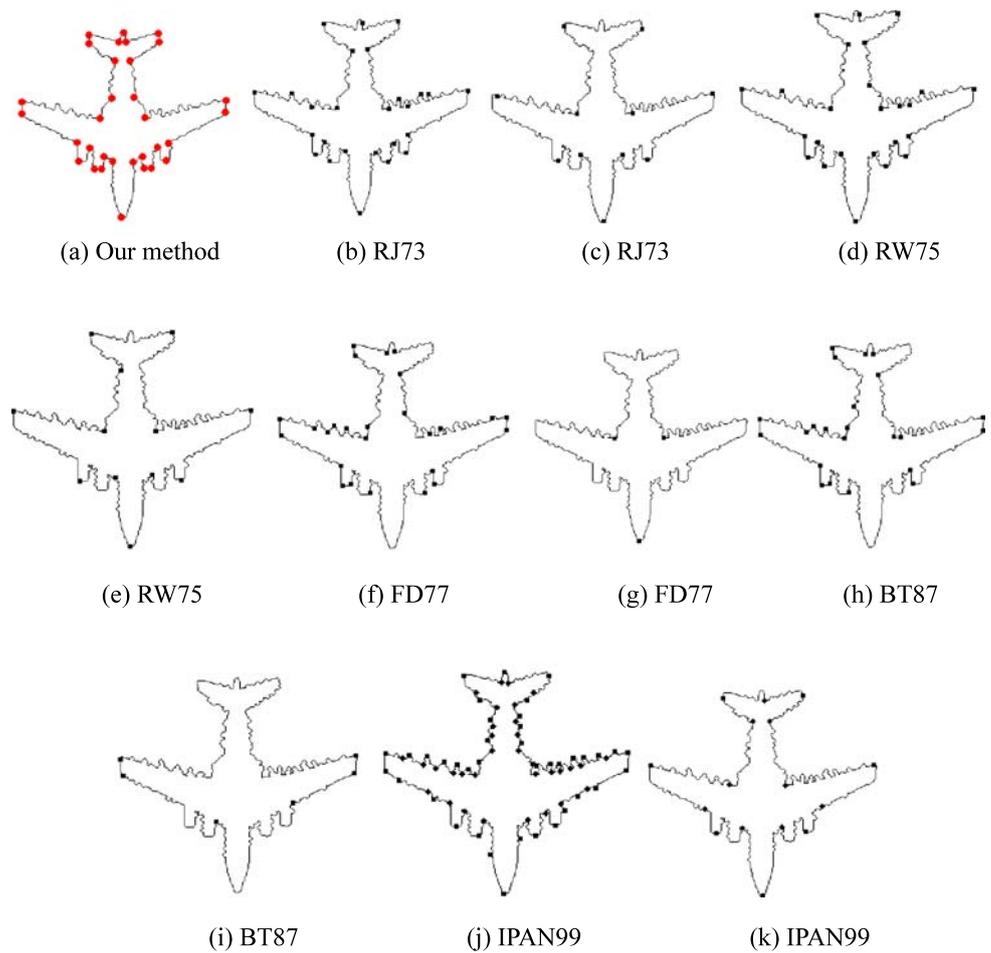
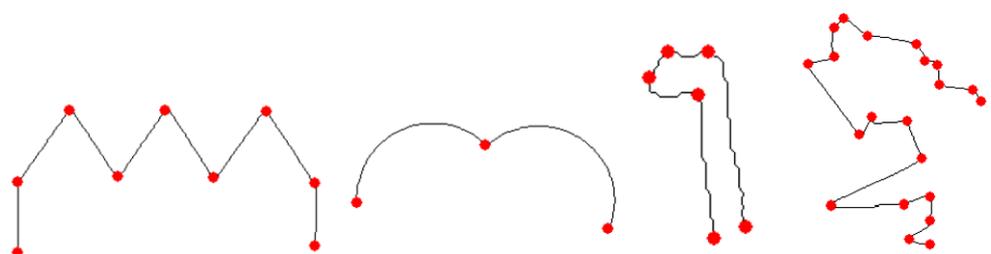


Fig. 29 The detected corners on open curves



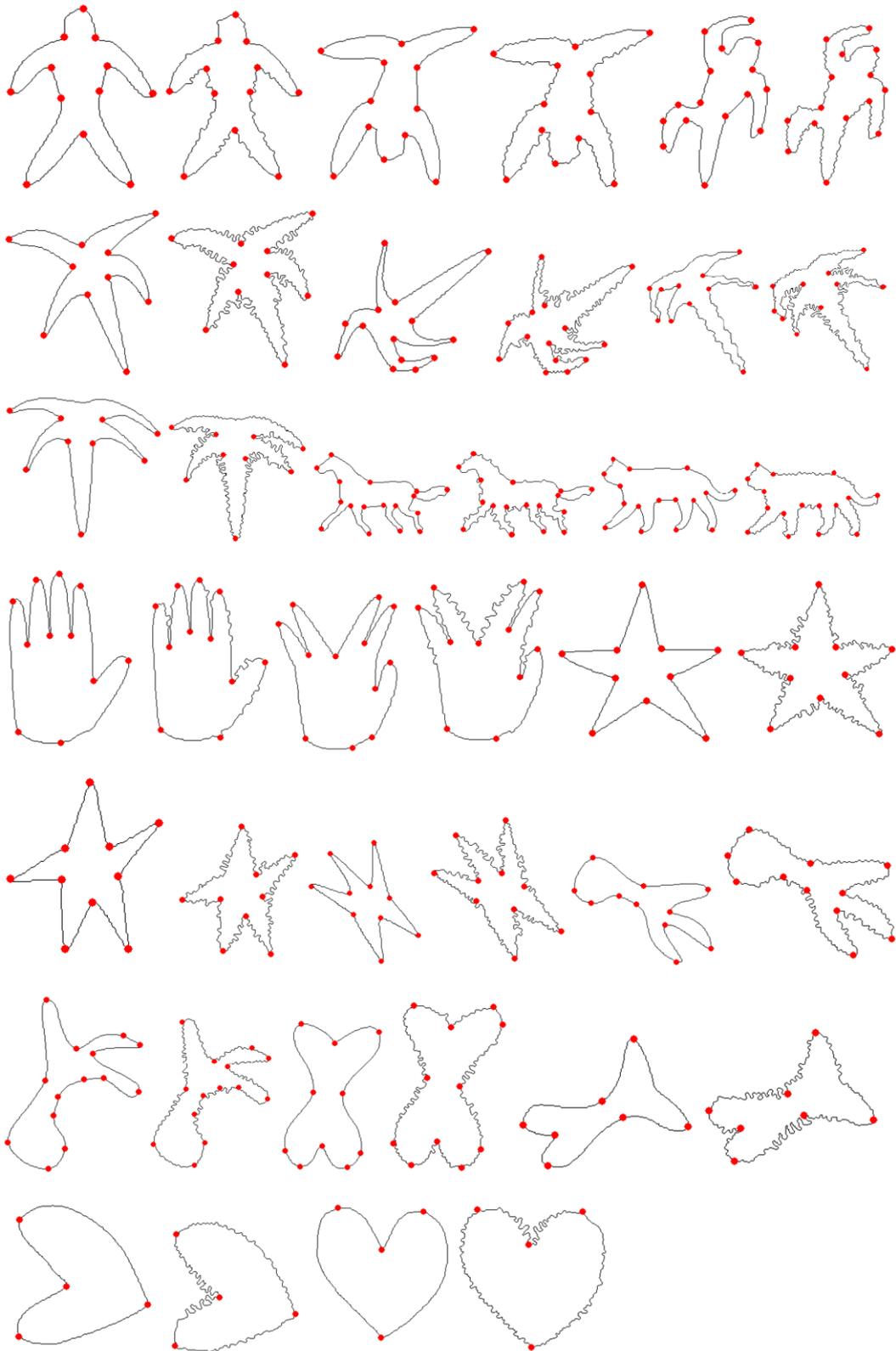


Fig. 30 Corner detection results on twenty articulated shapes from Asian and Tari (2005). For each shape we also show its copy with significant noise added

tors, it is hard to choose a proper scale, since their scale parameters are proportional to arc length. When they are large, many corners can not be detected; when they are small, many noisy points are detected as corners. For the purpose of comparison, for each detector, we illustrate in Fig. 28 the results on two scales, one scale is small and many true corners are detected, the other is large and many false corners caused by noise are eliminated. Table 1 shows the values of scale parameters for the experiments in Fig. 28.

Figure 28 shows that our corner detector works well even under significant noise. The other five corner detectors do not yield good results; they either have many false corners or miss some true corners.

To demonstrate the results of our method on open curves, we extracted contour parts of some in Fig. 20. The detected corners are shown in Fig. 29. In this experiment, we used the same parameters as for the complete shapes. According to our curvature definition, the curvatures of the endpoints of open curves are undefined, however, because they are very important, they are considered as corner points. When endpoints are ignored, all other corner points detected are identical to the corner points detected on corresponding parts of complete contours. This demonstrates the stability of the proposed approach.

To further demonstrate the stability of our method in the presence of articulation and noise, Fig. 30 demonstrates the corner detection result on 20 articulated shapes from Asian and Tari (2005). For each original shape, we also created its copy with significant contour noise added.

8 Conclusion

This paper proposes a new curvature definition which can be considered to be a geometric explanation of the standard curvature definition rooted in differential geometry. Based on this definition, a natural multi-scale curvature is introduced. In analogy with denoising method by “low pass” filter in signal theory, our multi-scale curvature can be considered to be a “low pass” filter. This analogy also explains why curvature is hard to be estimated accurately and robustly, because it usually contains small-scale signals caused by noise.

We proved for regular curves that the limit of visual curvature is the standard curvature, and that visual curvature is identical to turn angle at vertices for polygonal curves. Thus, standard curvature and turn angle are just two special cases of proposed visual curvature. In this way, we unify the definition of curvature on both regular curves and polygonal curves.

The properties of the multi-scale visual curvature are investigated and their practical significances are demonstrated. Especially, we discuss the geometric meaning of our scale measure. Since the scale measure is defined in global, multi-scale visual curvature is in fact estimated globally.

Finally, we discussed the application of multi-scale visual curvature in corner detection, and show that using multi-scale visual curvature, we can detect intuitive corners of various shapes at different scales robustly.

Acknowledgement This work is supported by Doctoral Fund of Ministry of Education of China (NO: 20070487028), and the Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China (NO: 705038). This work was also supported by the NSF under Grant No. IIS-0534929 and by the DOE under Grant No. DE-FG52-06NA27508.

References

- Adamek, T., & Connor, O. (2004). A multiscale representation method for nonrigid shapes with a single closed contour. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5), 742–753.
- Anderson, M., & Bezdek, J. C. (1984). Curvature and tangential deflection of discrete arcs: A theory based on the commutator of scatter matrix pairs and its application to vertex detection in planar shape data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 27–40.
- Ansari, N., & Delp, E. J. (1991). On detecting dominant points. *Pattern Recognition*, 24(5), 441–451.
- Asada, H., & Brady, M. (1986). The curvature primal sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8, 2–14.
- Asian, C., & Tari, S. (2005). An axis-based representation for recognition. In *IEEE international conference on computer vision* (Vol. 2, pp. 1339–1346).
- Bai, X., Latecki, L. J., & Liu, W.-Y. (2007). Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 29, 449–462.
- Belyaev, A. (2004). *Plane and space curves. Curvature. Curvature-based features*. Max-Planck-Institut für Informatik.
- Bengtsson, A., & Eklundh, J.-O. (1991). Shape representation by multiscale contour approximation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1).
- Beus, H. L., & Tiu, S. S. H. (1987). An improved corner detection algorithm based on chain-coded plane curves. *Pattern Recognition*, 20, 291–296.
- Boutin, M. (2000). Numerically invariant signature curves. *International Journal of Computer Vision*, 40, 235–248.
- Calabi, E., Olver, P. J., Shakiban, C., Tannenbaum, A., & Haker, S. (1998). Differential and numerically invariant signature curves applied to object recognition. *International Journal of Computer Vision*, 26, 107–135.
- Cartan, E. (1935). *Lamethode du repere mobile, la theorie des groupes continus et les espaces generalises exposes de geometrie No. 5*. Paris: Hermann.
- Cederberg, R. L. T. (1978). An iterative algorithm for angle detection on digital curves. In *International conference on pattern recognition* (pp. 576–578). Kyoto, Japan.
- Chetverikov, D., & Szabo, Zs. (1999). A simple and efficient algorithm for detection of high curvature points in planar curves. In *Proceedings of 23rd workshop of the Austrian pattern recognition group* (pp. 175–184).
- Coerjolly, D., Serge, M., & Laure, T. (2001). Discrete curvature based on osculating circles estimation. In *Lecture notes in computer science* (Vol. 2059, pp. 303–312). Berlin: Springer.
- Cong, G., & Parvin, B. (1998). Curve evolution for corner enhancement. In *International conference on pattern recognition* (Vol. 1, pp. 708–710).

- Dudek, G., & Tsotsos, J. K. (1997). Shape representation and recognition from multiscale curvature. *Computer Vision and Image Understanding*, 68(2), 170–189.
- Dunham, J. G. (1986). Optimum uniform piecewise linear approximation of planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8, 67–75.
- Freeman, H., & Davis, L. S. (1977). A corner-finding algorithm for chain-coded curves. *IEEE Transactions on Computers C*, 26, 297–303.
- Gumhold, S. (2004). *Designing optimal curves in 2D*. CEIG 2004.
- Hermann, S., & Klette, R. (2003). Multigrid analysis of curvature estimators. In *Image vision and computing*, New Zealand (pp. 108–112).
- Hermann, S., & Klette, R. (2005). Global curvature estimation for corner detection. In *International conference on image and vision computing (IVCNZ)*, Dunedin/New Zealand (pp. 272–277).
- Hermann, S., & Klette, R. (2007). A comparative study on 2d curvature estimators. In *International conference on computing: Theory and applications (ICCTA'07)*.
- Katzir, N., Lindenbaum, M., & Porat, M. (1994). Curve segmentation under partial occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16, 513–519.
- Kovalevsky, V. (2001). Curvature in digital 2D images. *International Journal of Pattern Recognition and Artificial Intelligence*, 15, 1183–1200.
- Kruse, B., & Rao, C. V. K. (1978). A matched filtering technique for corner detection. In *International conference on pattern recognition* (pp. 642–644). Kyoto, Japan
- Latecki, L. J., & Rosenfeld, A. (1998). Supportedness and tameness differentialless geometry of plane curves. *Pattern Recognition*, 31(5), 607–622.
- Lewiner, T. (2004). Arc-length based curvature estimator. In *SIB-GRAPI'04* (pp. 250–257).
- Lewiner, T., Gomes, J., Jr., Lopes, H., & Craizer, M. (2004). *Curvature estimation: Theory and practice*. Prepublicacoes do Departamento de Matematica da PUC-Rio.
- Lowe, D. G. (1988). Organization of smooth image curves at multiple scales. In *International conference on computer vision* (pp. 558–567).
- Lu, F., & Milios, E. E. (1991). Optimal local spline approximation of planar shape. In *International conference on acoustics, speech, and signal processing* (Vol. 4, pp. 2469–2472).
- Marji, M. (2003). *On the detection of dominant points on digital planar curves*. Ph.D. Thesis, June 2003.
- Medioni, G., & Yasumoto, Y. (1986). Corner detection and curve representation using cubic B-splines. In *IEEE international conference on robotics and automation* (Vol. 3, pp. 764–769).
- Mokhtarian, F., & Mackworth, A. K. (1992). A theory of multi-scale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14, 789–805.
- Mokhtarian, F., & Suomela, R. (1998). Robust image corner detection through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12), 1376–1381.
- Pavlidis, T. (1980). Algorithms for shape analysis and waveforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2, 301–312.
- Pinheiro, A. M. G., Izquierdo, E., & Ghanhari, M. (2000). Shape matching using a curvature based polygonal approximation in scale-space. In *International conference on image processing* (Vol. 2, pp. 538–541).
- Rattarangsi, A., & Roland, T. (1992). Scale-based detection of corners of planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14, 430–449.
- Rosenfeld, A., & Johnston, E. (1973). Angle detection on digital curves. *IEEE Transactions on Computers C*, 22, 875–878.
- Rosenfeld, A., & Weszka, J. S. (1975). An improved method of angle detection on digital curves. *IEEE Transactions on Computers C*, 24, 940–941.
- Sankar, P. V., & Sharma, C. V. (1978). A parallel procedure for the detection of dominant points on a digital curve. *Computer Graphics and Image Processing*, 7, 403–412.
- Utcke, S. (2003). Error-bounds on curvature estimation. In *Lecture note in computer science* (Vol. 2695, pp. 657–666). Berlin: Springer.
- Wang, Y.-P., Lee, S. L., & Toraiichi, K. (1999). Multiscale curvature-based shape representation using B-spline wavelets. *IEEE Transactions on Image Processing*, 8(11).
- Worring, M., & Smeulders, A. W. M. (1993). Digital curvature estimation. *CVGIP: Image Understanding*, 58, 366–382.
- Yuille, A. L. (1989). Zero crossings on lines of curvature. *Computer Vision Graphics Image Process*, 45, 68–87.