A new iterative approach for dominant points extraction in planar curves

CECILIA DI RUBERTO
University of Cagliari
Dept. of Mathematics and Comp. Science
Via Ospedale, 72 09124 Cagliari
ITALY
dirubert@unica.it

ANDREA MORGERA
University of Cagliari
Dept. of Mathematics and Comp. Science
Via Ospedale, 72 09124 Cagliari
ITALY
andrea.morgera@unica.it

Abstract: In this paper the problem of dominant point detection on digital curves is addressed. Based on an initial set of curvature points, our approach adds iteratively significant points by looking for the higher curvature contour points. The process continues until all the sums of the distances of contour points in the arcs subtended to the chord between two next dominant points is less then a predefined threshold. A final refinement process adjusts the position of located dominant points by a minimum integral square error criterion. We test our method by comparing its performance with other well known dominant point extraction techniques succesfully. In the last section some examples of polygonal approximation are shown.

Key–Words: Curvature, Digital Curve, Dominant Points, Polygonal Approximation.

1 Introduction

An important goal in image understanding is to detect, track and label objects of interest present in observed images [1, 2, 3]. Image objects can be characterized in many ways according to their colors, textures, shapes, movements and locations. In the field of machine vision applications, the aim of shape recognition is to identify an object correctly. There are two typical approaches to the problem : *global* or *internal* methods, which use all the points of the objects and contour or *external* methods which are focused on the analysis of contour points of the objects. Planar curves that represent contours of objects have been studied independently for a long time. An emerging opinion in the vision community is that global features such as shapes of contours should also be taken into account for the successful detection and recognition of objects. The term *dominant point* is assigned to points having a sufficiently high curvature on the boundary of a planar object: their detection is a very important aspect in contour methods since information on the shape of a curve is concentrated at dominant points. This representation simplifies the analysis of the images by providing a set of feature points containing almost complete information of a contour. One of the main advantages of a good representation of 2D shapes through dominant points is the high data reduction and its immediately efficiency in feature extraction and shape matching alghoritms. It is well known also that these points play a dominant role in shape perception by humans.

The problem of detecting points of high curvature in 2D shapes has been researched since the early 1970's [4, 5]. Most of these algorithms require one or more parameters that specify (directly or indirectly) the region of support in order to measure the local properties at each point of the curve.

Many methods have been developed to find dominant points. Basically they fall in two categories. The first one directly extracts dominant points from grey level images [6, 7, 8]. The second needs a prior edge detection to extract the contour and works afterwards on the chain code formed by the contour [1, 4, 9, 10, 11, 12]. This latest category corresponds to our field of investigation.

The purpose of this paper is to give a different approach to dominant points extraction. In this work we will focus on an algorithm that attempts to represent shapes with a limited number of dominant points located along their boundaries. We consider a set of points as a dominant set if is possibile to reconstruct the contour from these points using some method or interpolation (polygon is the simplest case). The method is based on a iterative approach. The first step is to assign an initial set of points and after detect other possible dominant points, i.e. relevant points, applying an iterative selection based on a particular distance criterion. The results we obtain are then refined by suppressing some overmuch points in critical regions of the shape contour and by positioning them in order to minimize the approximation error. The detection procedure is compared with methods proposed by [5], [11], and by [13]. These methods relies on the curvature estimation approach and on the determination of the region of support. In section 2 a description about curvature both in eculidean space and in discrete space is given. The Freeman coding for a digital curve it is also described. In section 3 we expose our method. In section 3.1 the process to locate initial points set for the iterative method is explained. Section 3.2 describes the algorithm with its main steps. In section 3.3 and 3.4 a refinement technique for a better localization of dominant points is proposed. The experimental results are shown in section 4 and finally the main conclusions are summarized in section 5.

2 On dominant point detection

Since observing that information of a curve is concentrated on extreme curvature points, corner detection is an important research area in contour methods of shape analisys. For this reasons these points are often called *dominant points*. In the real Euclidean plane, curvature is defined as the rate of change between the tangent angle and the arc length:

$$k = \frac{d\theta}{ds}$$

For a curve $y = f(x)$ it can be also expressed in terms of derivatives as:

$$\frac{\frac{d^2y}{dx^2}}{\left[1+\left(\frac{dy}{dx}\right)^2\right]^{3/2}}$$

A closed digital curve can be describe by a sequence of *n* integer-coordinate points:

$$C = \{p_i = (x_i, y_i), i = 1, ..., n\}$$

where p_{i+1} is a neighbor of p_i (modulo n). A digital curve can also be represented using the Freeman's chain codes [14]. The Freeman chain code of C consists of the n vectors

$$\overrightarrow{c}_i = \overline{p_{i-1}p_i}$$

each corresponding to an integer :

$$f = 0, ..., 7$$

as shown in fig. 2 where 1/4πf is the angle between the X-axis and the vector. The chain of C is defined as $\{\overrightarrow{c}_i, i = 1, ..., n\}$ and $\overrightarrow{c}_i = \overrightarrow{c}_{i \pm n}$. If $c_{i-1} = c_i$, P_i is a linear point, otherwise it is a *break point* and is a candidate to dominant points.

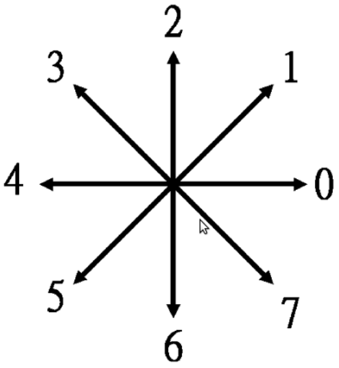

Figure 1: Freeman's chain code.

2.1 Previous related works

Although the notion of curvature for continuous curves is an unambiguously defined concept from differential calculus that admits precise analytical formulations, its digital counterpart lacks a universally accepted definition. As a result, research in this field has followed different, but conceptually related, approaches in order to circumvent the problems associated with digital curvature estimation. Many algorithms to find extreme points on digital curves have been suggested: one approach tried to define alternative curvature measures based on angles between vectors that connect the curve points [4, 9, 10, 11, 15, 16, 17]. Other techniques used curvature morphology [18, 19], local neighborhood of curve points in the plane [20, 21], arc-chord distance [22, 23, 24], curve fitting techniques [25, 26], local symmetry of shape [27], adaptive Gaussian smoothing [28], curvature-guided polygonal approximation [29], direct chain code analysis [30, 31], neural networks [32, 33], deviation from smoothness [34], wavelets [35, 36], electrostatics [37], fuzzy logic [38], B-splines and genetica algorithms [39]. We focused our attention on three region of support based algorithms.

2.1.1 Teh-Chin method

Teh and Chin [11] used the ratio of distance d_{ik} between P_i and the chord $\overline{P_{i-k}, P_{i+k}}$ (denoted with l_{ik}) to determine the region of support of a candidate dominant point :

$$r_{ik} = \frac{l_{ik}}{d_{ik}}$$

To estimate the length of support region, the process starts with k = 1, increases k by one each until one of the following conditions is satisfied, then k determines the length of the support region at point P_i:

$$\star \ l_{ik} \geq l_{ik+1}$$

$$\star \ \begin{cases} r_{ik} \geq r_{ik+1} \ \ for \ \ d_{ik} > 0 \\ r_{ik} \leq r_{ik+1} \ \ for \ \ d_{ik} < 0 \end{cases}$$

This ratio can be considered a measure equivalent to curvature.
After calculating for each point P_i its support region, a *significance measure* $|S(P_i)|$ is given by one of the following definitions:

- *k-cosine* measure by Rosenfeld-Johnston [4]
- $CUR_{ik} = \frac{1}{k}\sum_{j=-k}^{-1} f_{i-j} - \frac{1}{k}\sum_{j=0}^{k-1} f_{i-j}$ [40] where f is one of the possible directions of Freeman coding
- 1-curvature $CUR_{i1} = f_{i+1} - fi$

At this point, a nonmaxima suppression of $|S(P_i)|$ is done by retaining only those points where $|S(P_i)| \geq |S(P_j)|$ for all j such that $|i - j| \leq k_i/2$.
A sort of ”fine tuning” adjustment in the position of some dominant points is made by suppressing those points having $CUR_{i1} = 0$. Then a further refinement by suppressing those points with $|S(P_i)|$ minimum is done. In [12] it is pointed out that Teh and Chin's algorithm is not robust to noisy contours, due to the fact that the local maximum curvature may be caused by noisy variations on the curve.

2.1.2 Wu methods

Wu [13] proposed a simple measurement to detect corners. He used an *adaptive bending value* to determine the region of support of each point in the contour. For all break points (described in sect. 2) the bending value is defined as:

$$b_{ik} = max(|(x_{i-k} - x_i) + (x_{i+k} - x_i)|, |(y_{i-k} - y_i) + (y_{i+k} - y_i)|)$$

A support region for the ith break point can be estimated by the following rule: start with k=1, compute the bending value b_{ik}, increase k by one each until one the following conditions is satisfied:

- If $b_{ik} \leq b_{ik+1}$ then k is the length of region of supports for P_i
- Else if $b_{ik} = b_{ik+1}$ then the greatest k that satisfies this condition is the length of regions of support for P_i

The estimated curvature in the point P_i can be obtained by the following smoothing bending value:

$$bv_i = \frac{1}{k_i}\sum_{j=1}^{k_i} b_{ij}$$

To eliminate the break points from the candidates to dominant points the following conditions are used:

- $bv_i < \epsilon$ where ϵ is a preset threshold;
- $bv_i < bv_j$ for j=i-1 or j=i+1;
- $bv_i = bv_{i-1}$ and $k_i < k_{i-1}$;
- $bv_i = bv_{i+1}$ and $k_i < k_{i-1}$;
- $bv_i = bv_{i+1}$ and $k_i = k_{i-1}$.

The remaining break points with local maximum over its support region are the dominant points.
With another approach [5] Wu proposed a different method for dominant points detection. After a first step, in which the support region identification for each break point P_i is similar to the Teh and Chin approach, the curvatures of all of the break points have been calculated by using the k-cosine measure. Then the points that have local maximum curvature are located. In addition, the points whose curvature is less than a predefined threshold are removed from the set of candidates for dominant points.

3 Proposed algorithm

3.1 Initial point setup

In order to reach a complete set of dominant points of a given a contour, we need of a initial group of starting points. The four vertex theorem [41] stats that a simple closed curve in the plane, other than a circle, must have at least four “vertices” that is at least four points where the curvature has a local maximum or a local minimum. By the result of this theorem we choose to locate four initial points by looking for local maxima and minima in the signature of the shape.
At first we calculate the centroid of the countour defined as:

$$x_c = \frac{1}{n}\sum_{i=1}^{n} x_i \quad , \quad y_c = \frac{1}{n}\sum_{i=1}^{n} y_i$$

where (x_i, y_i) are the coordinates of generic P_i of the n contour points. Then we calculate the signature of the contour obtained as:

$$s_i = \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2}, i = 1, 2, ..., n$$

Given the signature profile of the contour we then calculate local maxima and local minima of this function. The values are computed by the following criterion: a point is considered a maximum peak if it has the maximal value, and was preceded (to the left) by a value lower by Δ. In our case we use Δ = 0.5. In such

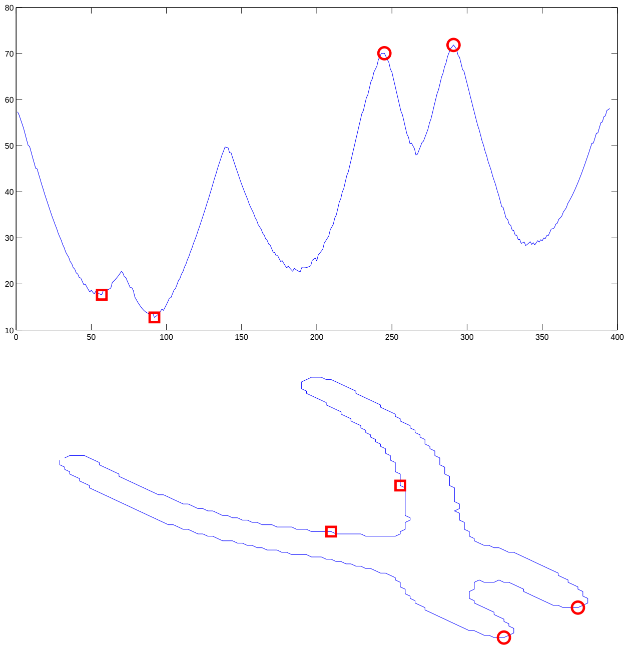

Figure 2: Signature for punch shape. Circle denote local maxima of curvature and square denote local minima of curvature.

way, searching the minimal values, we find minimum peaks. We use the first and second maximum and minimum peaks so we have four starting points for our method.

3.2 Iterative dominant point addition

After the initial points setup the method builds a set of dominant points in the following way:

1) Let $D = \{d_i, \quad i = 1, ..., m\}$ the intial dominant points set. In the first iteration m = 4 according to the starting points setup. For each pair of points $d_i d_{i+1}, \quad i = 1, 2, ...m$ where d_{i+1} is a neighbor of d_i (modulo m) it is possible to calculate the sum of distance from each contour point $p_j \quad (j = 1, ..., n)$ between d_i and d_{i+1} and the line that connects the two points:

$$dist_j = \frac{|(x_{i+1} - x_i)(y_i - y_j) - (x_i - x_j)(y_{i+1} - y_i)|}{\sqrt{((x_{i+1} - x_i))^2 + ((y_{i+1} - y_i))^2}} \quad (1)$$

where (x_{i+1}, y_{i+1}), (x_i, y_i) and (x_j, y_j) are coordinates of d_{i+1}, d_i and p_j respectively. A global distance is calculated as :

$$T = \sum_j dist_j \quad , \quad j = 1, ..., n$$

2) In order to find dominant points the value T of the previous step is compared to a treshold s defined as $s = p * l$, where p is an input parameter of the algorithm and l is the number of points of the line that connects d_i and d_{i+1}. If $T > s$ we add an intermediate point between d_i and d_{i+1} looking for the point with maximum distance to the chord. The current set of points and the new dominant points added are the set of points for the next iteration.

3) The algorithm stops when, after two following iterations, the number of the current set of points is equal to the number of the previous set, i.e. each distance T is less or equal to the corresponding threshold s.

3.3 Overmuch point suppression

Given a dominant points set D we adopt a refinement technique to suppress dominant point which are near enough between them. Let $l_{i,j}$ the distance between d_i and d_j and τ a given threshold, if $l_{i,j} < \tau$ then we substitute $d_i \equiv (x_i, yi)$ and $d_{i+1} \equiv (x_{i+1}, y_{i+1})$ with their meidum point :

$$\overline{d_k} \equiv (\overline{x_k}, \overline{y_k})$$

where $\overline{x_k} = \frac{x_i + x_{i+1}}{2}$ and $\overline{y_k} = \frac{y_i + y_{i+1}}{2}$. A reasonable value of τ = 0.007 is set as default.

3.4 Point localization arrangement

Once dominant points are detected we apply an enhancement process to refine results adjusting this point configuration according to a minimal distance criterion.

Let $\{d_1, d_2, ..., d_M\}$ the final ordered set of dominant points. Considering the couples d_i and $d_{(i+2)modM}$ ($i = 1, ..., M$), the improvement method consists in moving an intermediate point between d_i and $d_{(i+2)}$ by choosing it in the arc $\widehat{c_k c_w}$, where $c_z, (z = k + 1, ..., w - 1)$ is a generic point of original contour between $c_k \equiv d_i$ and $c_w \equiv d_{(i+2)}$, so that a local minimization of distance is achieved.

The process acts as follows :

- for each pair d_i and $d_{(i+2)modM}$, $i = 1, ..., M$ the point c_z is choosen along the arc $\widehat{c_k c_w}$ described above;
- each time the quantity

$$DIST_{[i,z,(i+2)]} = DIST_{i,z} + DIST_{z,(i+2)}$$

where

$$DIST_{i,z} = \sum_{c_k \in \widehat{c_i c_z}} dist(c_k, \overline{c_i c_z})$$

and

$$DIST_{z,(i+2)} = \sum_{c_k \in c_z \widehat{c_{(i+2)}}} dist(c_k, \overline{c_z c_{(i+2)}})$$

as described in (1), is evaluated;

- the new index $(i+1)^*$ that minimize this sum is choosen :

$$(i+1)^* = \min_z DIST_{[i,z,(i+2)]}$$

- the new dominant point $d_{(i+1)^*}$ between d_i and $d_{(i+2)}$ and it is then used in the rest of the process.

4 Experimental results

In order to test the effectiveness of our method we perform some experiments, both on typical shapes (introduced by [4] and [11] and commonly used in many studies) and on biological and common tools shapes. In particular we test four contour commonly used curves : chromosome, infinity, leaf and semicirceles curves. Their contour and the set of dominant points, obtained by each method and highlighted with a small circle, are shown in figs. 3 - 6. Our method is compared with others by showing results with the same or nearest number of dominant points.

The proposed method is compared with the other methods and results are shown in table 1. We report some meaningful measures for each tested method: the number of dominant points *m*, the compression ratio (*CR*), the E_2 error and the E_2/CR ratio. *CR* is defined as the ratio between the number of dominant points *m* and the number of contour points *n*. The E_2 measure (also known as *Integral Square Error*) is tipically used to evaluate the effectiveness of polygonal approximation. It is defined as:

$$E_2 = \sum_j d_j^2 \quad , \quad j = 1, ..., n$$

where d_j is the distance of a contour point p_j , $j = 1, ..., n$ from the segment between p_i^* and p_{i+1}^* , p_i^* , $i = 1, ..., m$ is the i-th of dominant points such that $p_i^* \leq p_l \leq p_{i+1}^*$. We report the value of threshold *s* choosen also. The results clearly demonstrate better perfomance of our method respect to other techniques. Infact the E_2 error (and subsequently the E_2/CR value) is always lower keeping the same number of dominant points of other methods. Viceversa if we set an acceptable value of E_2 error our method gives the lowest number of dominant points, i.e. well-located points, compared to the other. After the refinement process described in 3.4 the E_2 error decrease some more. The comparison of results obtianed applying or not this step are shown in table 2.

We test our method and compare it to the other dominant points extraction techniques on common shapes (taken by [42] and [43]) also. In fig. 7 it's possibile to see some shapes we used in experiments. These images are processed by tresholding camera taken objects and are affected by noise. For each image the graph which shows the E_2 error trend against the dominant point number is plotted. We use the log scale for the error E_2. We obtain such values by varying the input parameter for each method except for the Teh-Chin method which does not require any. These graphs confirm the effectiveness of our method. Worse results of other methods probably are due to the noise sensitivity.

The polygonal approximation for some real images like Africa's map and maple leaf (taken by [44]) are shown in fig. 8 and 9, respectively. We work with Matlab 2008 running on a Linux 64 bit operating system on AMD64 6000 platform. The Africa's map contour is composed by 1364 points: we calculate the approximation with 20, 30 and 40 points respectively (the dotted line is the original contour). The resulting computational time is 0.0756 sec., 0.0731 sec and 0.0659 sec. The maple leaf shape is composed by 1630 points : we calculate the approximation with 40, 45 and 50 points respectively (the dotted line is the original contour). The resulting computational time is 0.1343 sec., 0.1449 sec and 0.1895 sec. respectively.

5 Conclusions

The problem to find a fast and accurate method to find a set of dominant points given a contour shape is addressed in this paper. The proposed method not only has a very low computational time and robustness to noise but also produces a good polygonal approximation while keeping low the E_2 error. By empirical evidence we notice that a threshold value in a range around 0.5 give the best E_2/CR value and an improvement of the proposed method by doing without the threshold input parameter in future works is planned.

References:

[1] V. Papić, H. Dujmić, ”A curve matching algorithm for dynamic image sequences”, WSEAS Transactions on Computers, vol. 4 (11), pp. 1713-1717, 2005

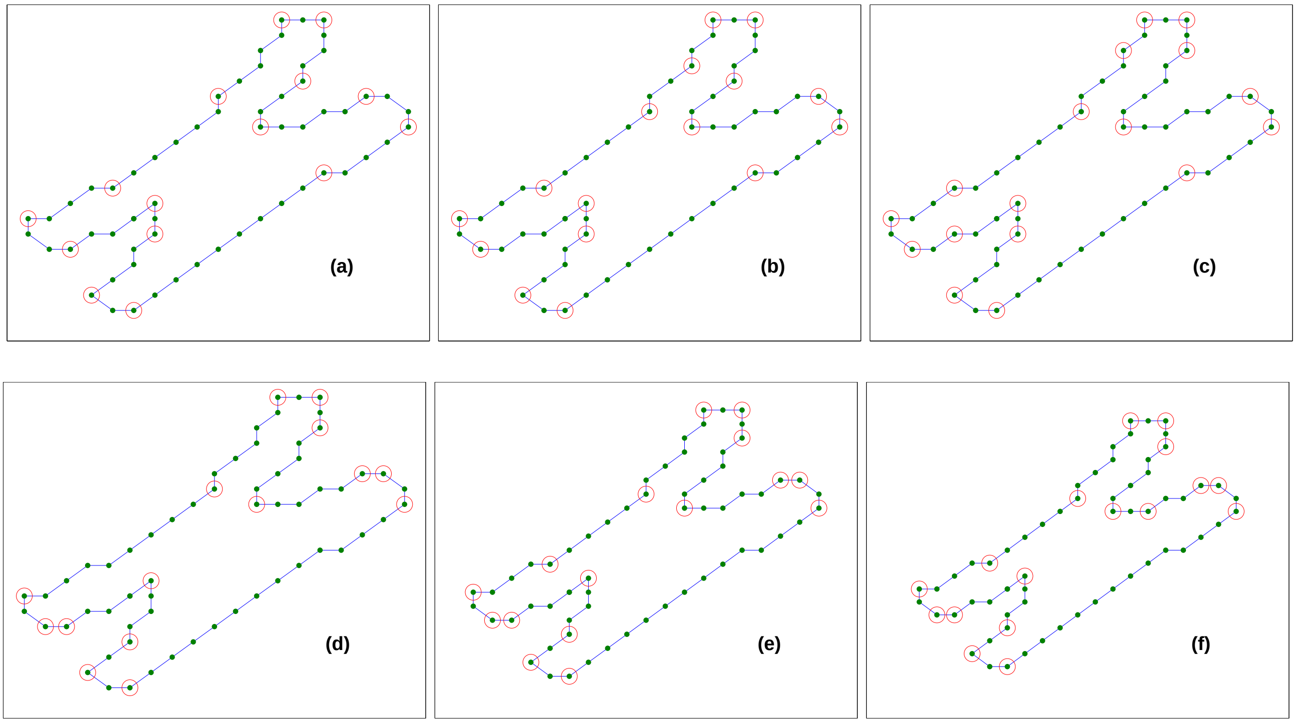

Figure 3: Chromosome shape (*50 points*) : (a) Teh-Chin (*15 points*), (b) Wu BV (*16 points*), (c) Wu DYN (*17 points*), (d) our method (*15 points*), (e) our method (*16 points*), (f) our method (*17 points*)

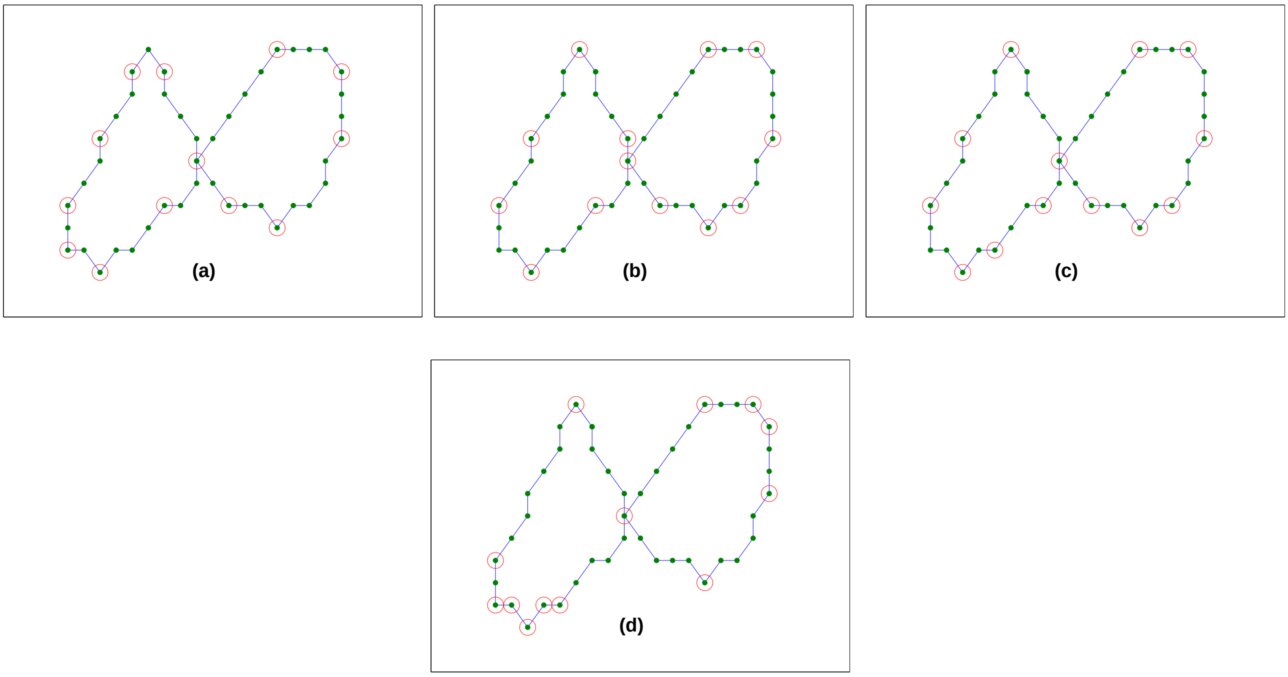

Figure 4: Infinite shape (*45 points*) : (a) Teh-Chin (*13 points*), (b) Wu BV (*13 points*), (c) Wu DYN (*13 points*), (d) our method (*13 points*)

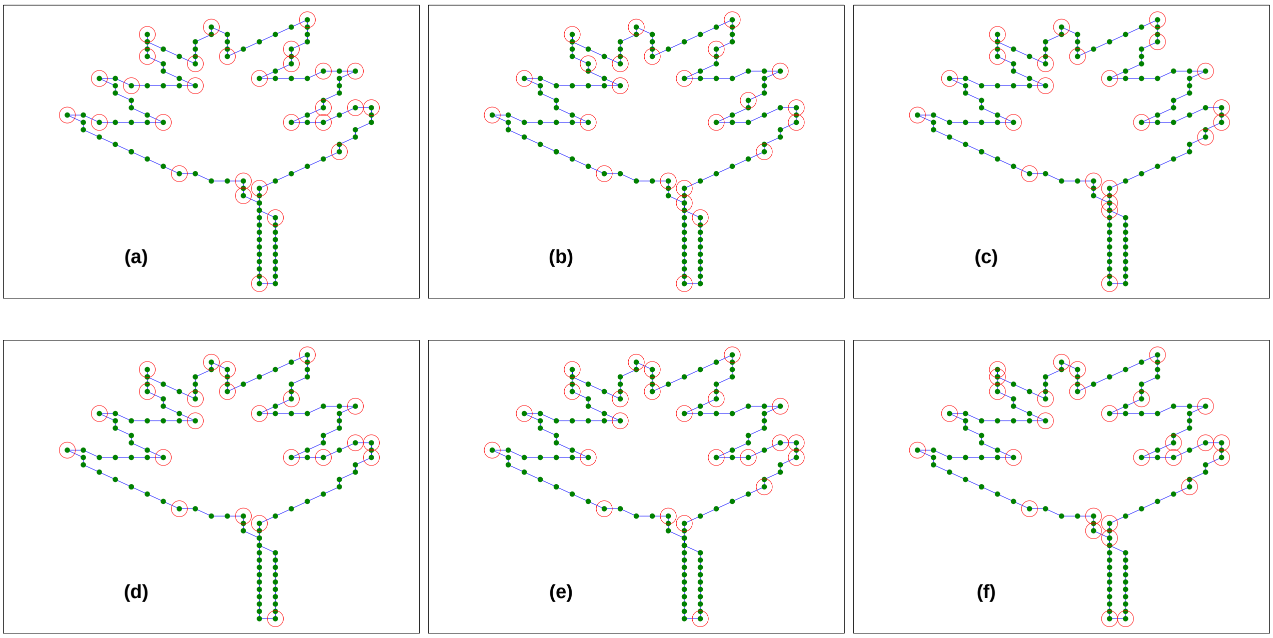

Figure 5: Leaf shape (*120 points*) : (a) Teh-Chin (*29 points*), (b) Wu BV (*24 points*), (c) Wu DYN(*23 points*), (d) our method (*23 points*), (e) our method (*24 points*), (f) our method (*29 points*)

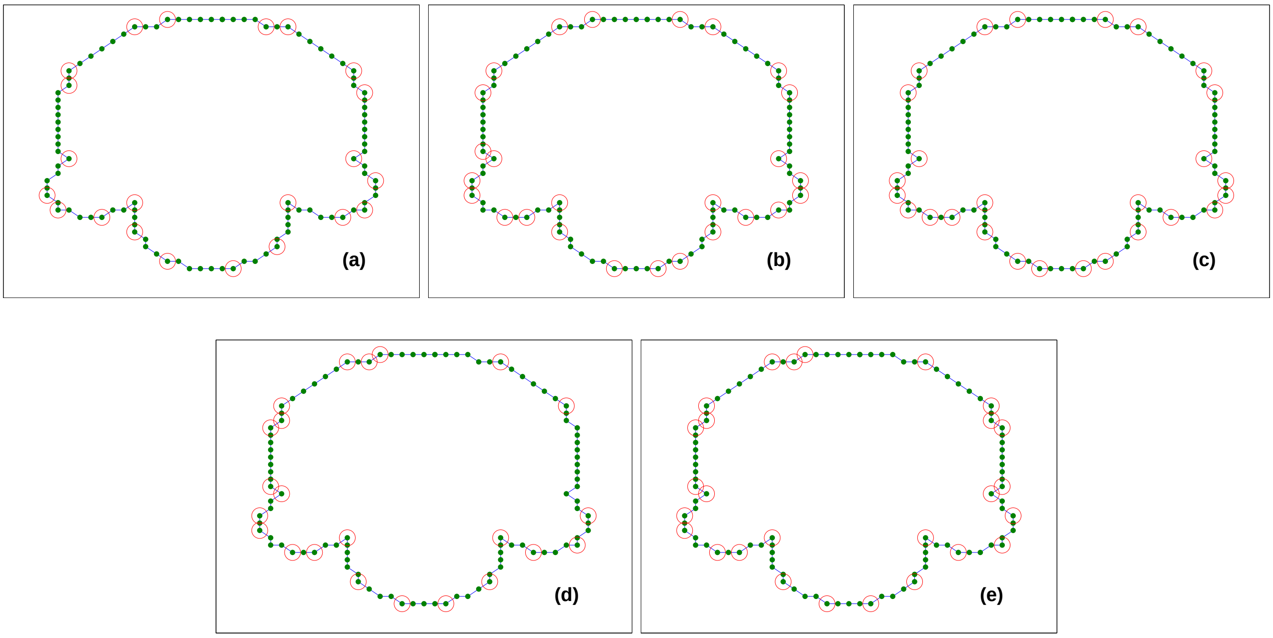

Figure 6: Semicircle shape (*102 points*) : (a) Teh-Chin (*22 points*), (b) Wu BV (*26 points*), (c) Wu DYN (*27 points*), (d) our method (*23 points*), (e) our method (*27 points*)

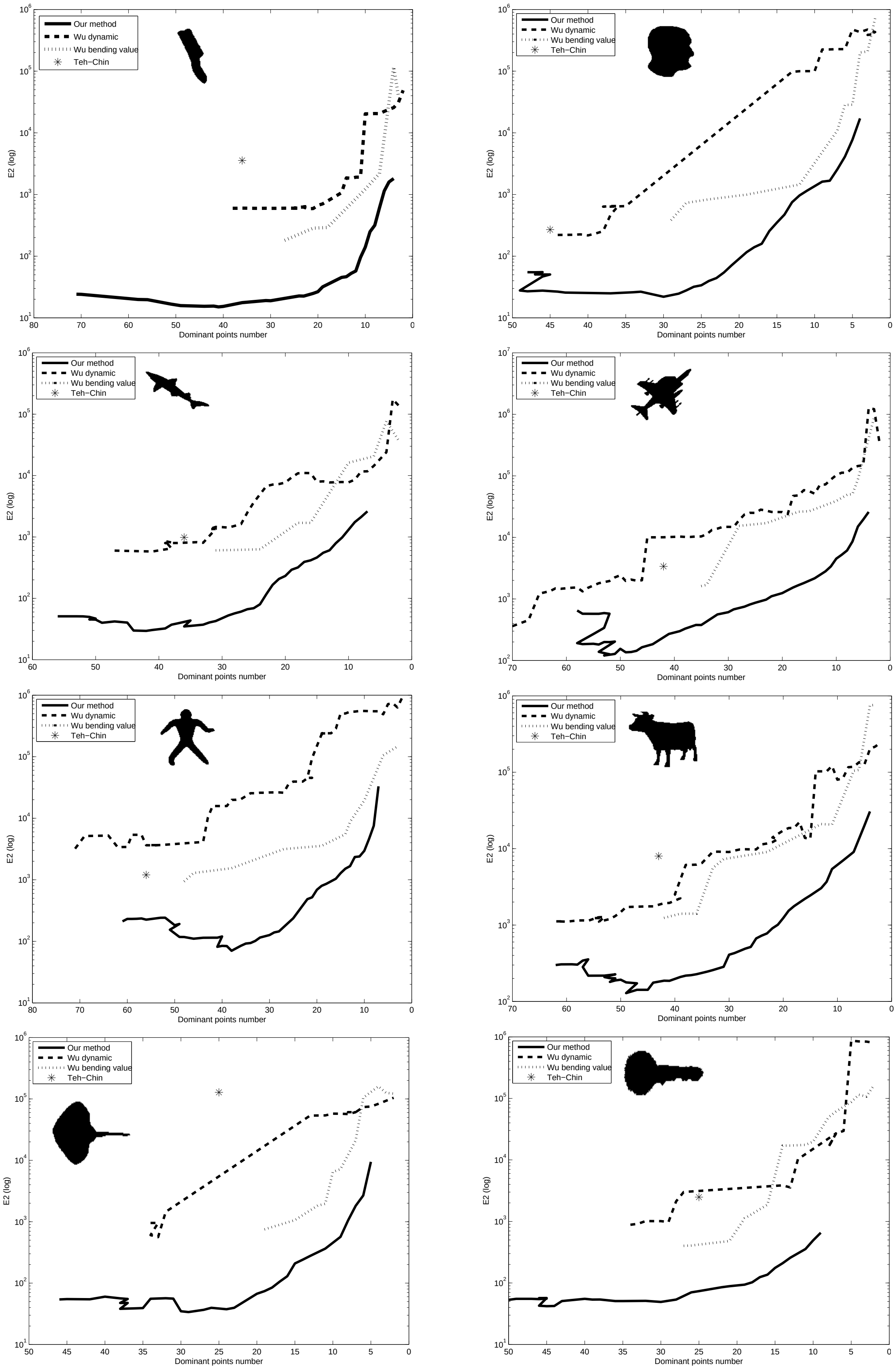

Figure 7: Shapes used in the experiments.

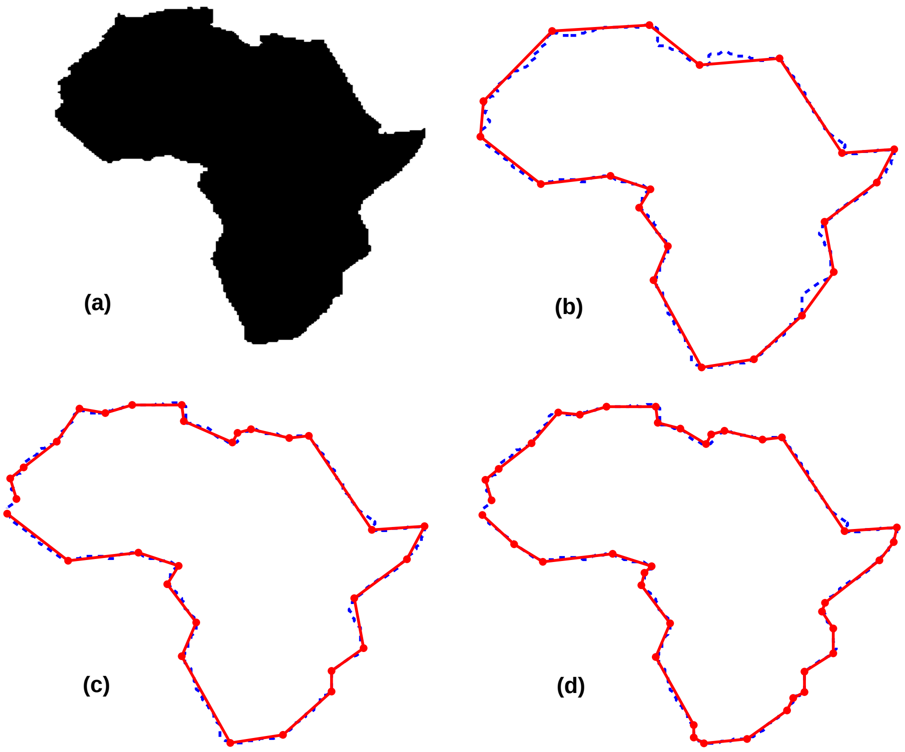

Figure 8: Africa's map : (a) original shape, (b) 20 points approximation, (c) 30 points approximation, (b) 40 points approximation

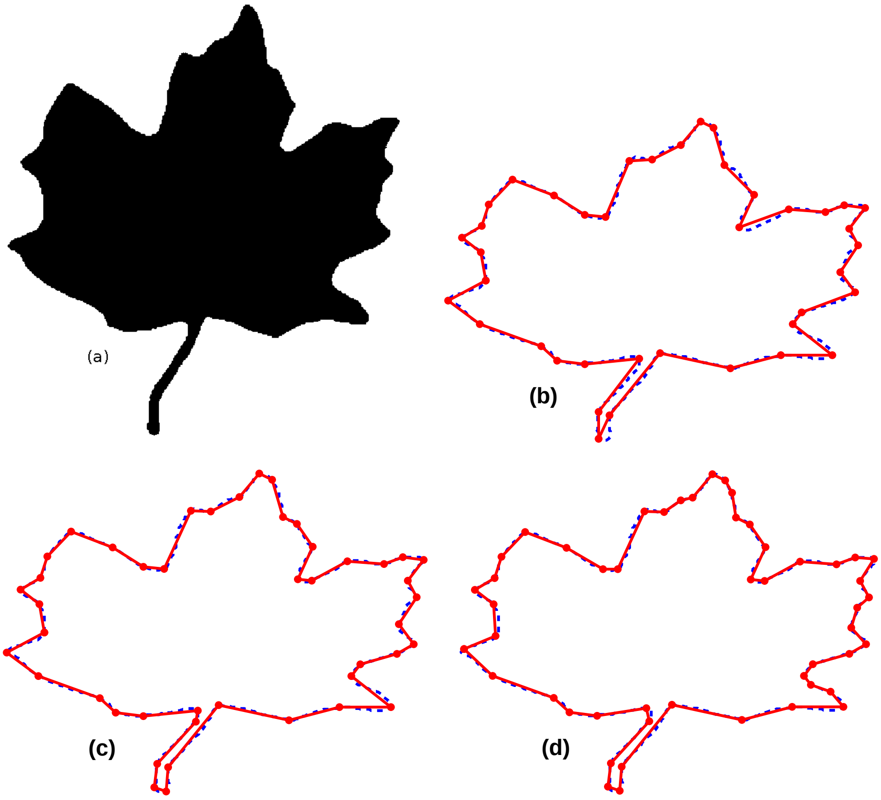

Figure 9: Maple leaf : (a) original shape, (b) 40 points approximation, (c) 45 points approximation, (b) 50 points approximation

Contour	Method	m	CR	E_2	s	E_2/CR
Chromosome (*points*=60)	Our method	15	4	6.5	.5	1.64
	Our method	16	3.75	4.62	.475	1.23
	Our method	17	3.53	3.74	.45	1.06
	Teh-Chin	15	4	7.19	none	1.8
	Wu_bv	16	3.75	4.70	n.a.	1.25
	Wu_dyn	17	3.53	5.01	n.a.	1,42
Leaf (*points*=120)	Our method	23	5.22	14.32	.45	2.74
	Our method	24	5	12.73	.42	2.55
	Our method	29	4.14	7.90	.365	1.91
	Teh-Chin	29	4.14	14.96	none	3.62
	Wu_bv	24	5	17.40	n.a.	3.48
	Wu_dyn	23	5.22	22.44	n.a.	4.3
Semicircle (*points*=102)	Our method	23	4.43	11.65	.4487	2.37
	Our method	27	3.78	6.85	.44	1.51
	Teh-Chin	22	4.64	21.86	none	4.45
	Wu_bv	26	3.92	9.04	n.a.	2.18
	Wu_dyn	27	3.78	9.92	n.a.	2.49
Infinite (*points*=45)	Our method	13	3.46	3.65	.4	1.05
	Teh-Chin	13	3.46	5.93	none	1.71
	Wu_bv	13	3.46	5.40	n.a.	1.56
	Wu_dyn	13	3.46	5.60	n.a.	1.62

Table 1: Comparative results for the proposed method

[2] V. Papić, H. Dujmić, ”A Procedure for Tracing Simple Curves in Image Sequences”, WSEAS Transactions on Information Science & Applications, vol. 3 (1), pp. 192-198, 2005

[3] C. Y. Liu and R. Du, ”Expert Label Pattern Design System”, WSEAS Transactions on Computers, Vol. 5 (9), pp. 2077-2083, 2006

[4] A. Rosenfeld, E. Johnston, ”Angle detection on digital curves”, *IEEE Trans. Comput.*, vol. C-22, pp. 875-878, Sept. 1973

[5] W. Y. Wu, ”A dynamic method for dominant point detection”, *Graphical Models*, vol. 64, pp. 304-315, 2003

[6] R. Mehrotra, S. Nichany, ”Corner detection”, Pattern Recognition vol. 23 pp. 12231233, 1990

[7] X. Xie, ”Corner detection by a cost minimization approach”, Pattern Recognition vol. 26 pp. 12351243, 1993.

[8] Z. Zheng, H. Wang, E.K. Teoh, ”Analysis of gray level corner detection”, Pattern Recognition Letters vol. 20 pp. 149162, 1999

[9] H. Freeman, L. S. Davis, ”A corner-finding algorithm for chain-coded curves”, *IEEE Trans. Comput.*, vol. C-26, pp. 297-303, Mar. 1977

[10] A. Rosenfeld, J. S. Weszka, ”An improved method of angle detection in digital curves”, *IEEE Trans. Comput.*, vol. C-24, pp. 940-941, Sept. 1975

[11] C. Teh, T. Chin, ”On the detection of. dominant points on digital curves”, *IEEE Trans. Pattern*

Contour	Without	With
Chromosome (15 pts.)	6.5	4.21
Chromosome (16 pts.)	4.62	3.78
Chromosome (17 pts.)	3.74	3.59
Leaf (23 pts.)	14.32	12.47
Leaf (24 pts.)	12.73	12.12
Leaf (29 pts.)	7.90	7.84
Semicircle (23 pts.)	11.65	9.14
Semicircle (27 pts.)	6.85	5.61
Infinite (13 pts.)	3.65	2.88

Table 2: E_2 comparative results for the proposed method with and without the final refinement process

Anal. Machine Intelligence, vol. 11, pp. 859-872, Aug. 1989

[12] P. Cornic, ”Another look at dominant point detection of digital curves”, Pattern Recognition Letters vol.18 pp-13-25, 1997

[13] W. Y. Wu, ”Dominant point detection using adaptive bending value”, *Image and Vision Comp.*, vol. 21, pp. 517-525, 2003

[14] H. Freeman, ”On the encoding of arbitrary geometric configurations”, IRE Transactions on Electronic Computers, vol. 10, pp. 260-268, 1961

[15] B. K. Ray, K. S. Ray, ”An algorithm for detection of dominant points and polygonal approximation of digitized curves”, *Pattern Rec. Letters*, vol. 13, N. 12, pp. 849-856, 1992

[16] B. K. Ray, K. S. Ray, ”Detection of significant points and polygonal approximation of digitized curves”, Pattern Recognition Letters, vol. 12, pp. 443-452, 1992

[17] D. Chetverikov, Z. Szabo, ”A simple and efficient algorithm for detection of high curvature points in planar curves”, Proc. of the 23rd Workshop of the Austrian Pattern Recognition Group, pp. 174-184, 1999

[18] F. Leymarie, M.D. Levine, ”Curvature Morphology”, TR-CIM-88-26, Center of Intelligent Machines, McGill University, Montreal, Canada, 1988

[19] X. Zhang, D. Zhao, ”A parallel algorithm for detecting dominant points on multiple digital curves”, Pattern Recognition, vol. 30 pp. 239244, 1997

[20] P.V. Sankar, C.V. Sharma, ”A parallel procedure for the detection of dominant points on a digital curve”, Comput. Graphics Image Process. vol. 7 pp. 403412, 1978

[21] K.H. Cheng, W.H. Hsu, ”Parallel algorithms for corner following on digital curves”, Pattern Recognition Letters Vol. 8 pp. 4753, 1988

[22] T.Y. Phillips, A. Rosenfeld, ”A method for curve partitioning using arc-chord distance”, Pattern Recognition Letters Vol. 5(4) pp. 285288, 1987

[23] M.A. Fischler, H.C. Wolf, ”Locating perceptually salient points on planar curves”, IEEE Trans. Pattern Anal. Mach. Intell. vol. 16 (2) pp. 113129, 1994.

[24] J.H. Han, T. Poston, ”Chord-to-point distance accumulation and planar curvature: a new approach to discrete curvature”, Pattern Recognition Letters Vol. 22 pp. 11331144, 2001

[25] G. Medioni, Y. Yasumoto, ”Corner detection and curve representation using cubic B-splines”, Comput. Vision Graphics Image Process. vol. 39 pp. 267278, 1987

[26] D.M. Tsai, M.F. Chen, ”Curve fitting approach for tangent angle and curvature measurement”, Pattern Recognition vol. 27 (5) pp. 699711, (1994)

[27] H. Ogawa, ”Corner detection on digital curves based on local symmetry of the shape”, Pattern Recognition vol. 22 pp. 351357, 1989

[28] N. Ansari, K.W. Huang, ”Non-parametric dominant point detection”, Pattern Recognition vol. 24 pp. 849862, 1991

[29] N. Ansari, J. Delp, ”On detecting dominant points”, Pattern Recognition vol. 24 pp. 414415, 1991

[30] D. Sarkar, ”A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves”, Pattern Recognition Letters vol. 14 pp. 959964, 1993

[31] J. Koplowitz, S. Plante, ”Corner detection for chain coded curves”, Pattern Recognition vol. 28 pp. 843852, 1995

[32] C.H. Chang, S.Y. Hwang, ”2D curve partitioning by Kohonen Feature Maps”, J. Visual Comm. Image Rep. vol. 5 (2) pp. 148155, 1994

[33] Du-Ming Tsai, ”Boundary-based corner detection using neural networks”, Pattern Recognition vol. 30 pp. 8597, 1997

[34] X.C. Pei, J.H. Horng, ”Corner point detection using nest moving averages”, Pattern Recognition vol. 27 pp. 15331537, 1994

[35] J.S. Lee, Y.N. Sun, C.H. Chen, C.T. Tsai, ”Wavelet based corner detection”, Pattern Recognition vol. 26 pp. 853865, 1993

[36] A. Quddus, M.M. Fahmy, ”Fast wavelet-based corner detection technique”, Electron. Lett. vol, 35 (4) 287288, pp. 1999

[37] K. Wu, M. Levine, ”2D shape segmentation: a new approach”, Pattern Recognition Letters vol. 7 pp. 133140, 1996

[38] L. Li, W. Chen, ”Corner detection and interpretation on planar curves using fuzzy reasoning”, IEEE Trans. PAMI vol. 21 (11) pp. 12041210, 1999

[39] P. Phokharatkul, S. Kamnuanchai, C. Kimpan, S. Phaiboon, ”Comparative Study of B-Spline and Cardinal Spline with Genetic Algorithm for Invariant Shape Object Recognition”, WSEAS Transactions on Computers, vol. 5 (3), pp.536-543, 2006

[40] F.C.A: Groan, P.W. Verbeek, ”Freeman-code probabilities of object boundary quantized contours”, Comput. Vision, Graphics, Image Processing, vol. 7 pp.391-402, 1978

[41] D. DeTurck, H. Gluck, D. Pomerleano, D. S. Vick, ”The four vertex theorem and its converse”, *Notice of the AMS*, vol. 54, n. 2, pp. 192-207, Feb. 2007

[42] http://www.cs.rutgers.edu/pub/sven/rutgers-tools/

[43] http://www.lems.brown.edu/vision/software/

[44] A. Masood, ”Dominant point detection by reverse polygonization of digital curves”, *Image and Vision Comp.*, vol. 26, pp. 702-715, 2008