A new thresholding approach for automatic generation of polygonal approximations ☆

N.L. Fernández-García*, L. Del-Moral Martínez, A. Carmona-Poyato, F. J. Madrid-Cuevas, R. Medina-Carnicer

Department of Computing and Numerical Analysis, Maimonides Institute for Biomedical Research (IMIBIC), University of Córdoba, Córdoba, Spain

Abstract

The present paper proposes a new algorithm for automatic generation of polygonal approximations of 2D closed contours based on a new thresholding method. The new proposal computes the significance level of the contour points using a new *symmetric* version of the well-known Ramer, Douglas - Peucker method, and then a new *Adaptive* method is applied to *threshold* the normalized significance level of the contour points to generate the polygonal approximation. The experiments have shown that the new algorithm has good performance for generating polygonal approximations of 2D closed contours. Futhermore, the new algorithm does not require any parameter to be tuned.

Keywords:
Digital planar curves, Polygonal approximation, Dominant points, Significance of contour points, Thresholding techniques, Object recognition, Assessment of polygonal approximations, Corner points, Automatic algorithm

☆This work has been developed with the support of the Research Projects called TIN2012-32952 and BROCA both financed by Economy and Competitiveness Ministry of Spain and FEDER.

*Corresponding author

Email address: `ma1fegan@uco.es` (N.L. Fernández-García)

1. Introduction

Shape representation by polygonal approximation has been extensively used for constructing a characteristic description of a boundary in the form of a series of straight lines. This representation is very popular due its simplicity, locality, generality, and compactness [17, 24, 40]. In a closed digital planar curve, most of the information is located at points of high curvature [3], which are used to obtain polygonal approximations. These points are known as *dominant points* and are an important target in many machine vision applications [39].

The polygonal approximation of closed digital planar curves, also called *contours*, can be formulated in two ways [13]:

- *Minimum-distortion problem* or $Min - \epsilon$ *problem*: these methods are based on a predefined number N of vertices and try to generate the *optimal* polygonal approximation with N vertices or dominant points (DP) so that its distance from the contour is minimal among all the approximations with N vertices.

- *Minimum-rate problem* or $Min - \#$ *problem*: these methods are based on a predefined error measure ϵ and try to generate the polygonal approximations, with the minimal number of vertices, that distant from the contour by no more than ϵ.

Many methods have been proposed to construct polygonal approximations of closed digital planar curves. These methods can be classified into two major categories:

- *Optimization approach*: these methods are based on an optimization criterion [1, 13, 28, 29, 34, 35].

- *Heuristic approach*: these methods construct the approximating polygon using some reasonable geometric or perceptual features of the contours [10]

  - *Parametric approach*: these methods take into account a parameter to generate the polygonal approximations [2, 4, 5, 7, 18, 21, 22, 23].

  - *Non-parametric approach*: these algorithms generate the polygonal approximations without using parameters [19, 20, 30, 37, 39].

The optimization methods have two drawbacks [10]: although they find good solutions, the optimum depends on the applied criterion and requires a very high computational complexity, which is not suitable for real time applications. On the contrary, the heuristic methods do not guarantee any kind of optimum; however, these algorithms can find reasonable polygonal approximations for real time applications.

Real images show irregular shapes and a lot of variability and different levels of detail. Only an algorithm working independently of input parameters, adjusting itself to each curve, could give satisfactory results if an efficient and reliable representation of the contours is desired [11]. The need for reliable and robust non-parametric algorithms that can be applied to different curves satisfactorily is still an issue of high importance [19].

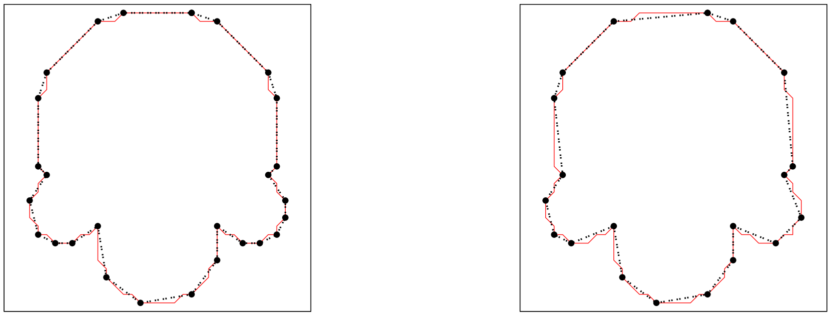
The present work proposes a new *automatic* and *non-parametric* method for polygonal approximations which is based on a new *symmetric* version of the well-known Ramer, Douglas - Peucker method [7, 31] and applies a new *Adaptive* thresholding method to obtain the dominant points. The present paper is arranged as follow. Section 2 describes a related work. Section 3 explains the new proposal. The experiments and results are described in section 4. Finally, the main conclusions are summarized in section 5.

2. Related work

The proposed algorithm is based on a modification of the Ramer, Douglas-Peucker method (RDP method)[7, 31], which is a scale-independent split algorithm for subdividing a curve into its most perceptually significant straight line segments. In order to use the RDP method to generate the polygonal approximation of a closed contour, the following steps can be applied:

1. Find the top-left and bottom-right points of the contour. These points are considered as *initial dominant points*.
2. The farthest point from the line segment defined by two consecutive dominant points is found [8]. The selected point is considered as new dominant point.
3. Repeat step 2 until the set of dominant points generates a polygonal approximation which fits the contour with a preset error.

The RDP method shows two drawbacks (see Figure 1): (1) it is a parametric method because the number of final dominant points (DP) depends on a preset error and (2) the symmetry of the contour is not taken into account. In order to avoid these drawbacks, the present work proposes a new automatic and non-parametric algorithm based on a modification of the RDP method which takes into account the symmetry of the contour (see section 3 and Figure 19(d)).



(a) Preset error: 0.7 (DP = 27)

(b) Preset error: 0.9 (DP = 20)

Figure 1: Drawbacks of the original RDP algorithm: dependence on a parametric error and lack of symmetry. The figure shows the polygonal approximations generated with two preset errors and the number of dominant points (DP).

3. Proposed algorithm to generate polygonal approximations

The proposed algorithm consists of six steps (see Figure 2):

1. *Selection of the initial points*: some special contour points are chosen to be considered *initial points*. See Section 3.1.
2. *Computation of the significance values of the non-initial points*: a *new* version of RDP algorithm [7, 31], which takes into account the symmetry of the contour, is used to compute the significance values of the non-initial points. This new symmetric version is described in Section 3.2.
3. *Computation of the significance values of the initial points*: the significance values of the *initial points* must be greater than the significance values of the other contour points (see Section 3.3).
4. *Generation of the normalized significance curve*: the significance curve must be normalized in order to facilitate the search for the dominant points using a thresholding method (see Section 3.4).

5. *Search for the threshold of the normalized significance values*: the normalized significance curve is used by a new *thresholding* method to search for the threshold of the normalized significance values. Four thresholding methods are proposed and described in Section 3.5. The performance of these thresholding methods has been analyzed in the first experiment (see Section 4.2) and the *Adaptive* method has obtained the best results and, therefore, this method has been chosen for this fifth step of the proposed algorithm.
6. *Thresholding of the normalized significance values to obtain the dominant points*: the normalized significance values of the contour points will be thresholded to obtain the dominant points of the contour. The points of the contour with normalized significance value equal to 1.0 or greater than the threshold will be chosen as *dominant points* (see Section 3.6).

In the best case, the computational complexity of the new algorithm is $\mathcal{O}(nlog(n))$; in the worst case, $\mathcal{O}(n^2)$ [9]. The following subsections 3.1-3.6 describe with more detail the six steps of the proposed algorithm to generate the polygonal approximation.

3.1. First step: selection of the initial points

The following method has been used to select the *initial points* of the polygonal approximation (see Figure 3):

- The farthest point(s) to the centroid is (are) chosen as *initial point(s)*.
- Besides, the farthest point(s) to the previous one(s) will be also considered as *initial point(s)*.

This method is independent of the starting point, invariant to rotations and scales and takes into account the symmetry of the contour. The computational complexity of this step is $\mathcal{O}(n)$.

3.2. Second step: computation of the significance of the non-initial points

The proposed method to compute the significance of the *non-initial* points modifies the original RDP method [7, 31] to take into account the symmetry of the contour. The new version consists of the following steps:

1. Let $I = \{P_1, \ldots, P_n\}$ be the set of *initial points* obtained by the method described in the first step (see Section 3.1):

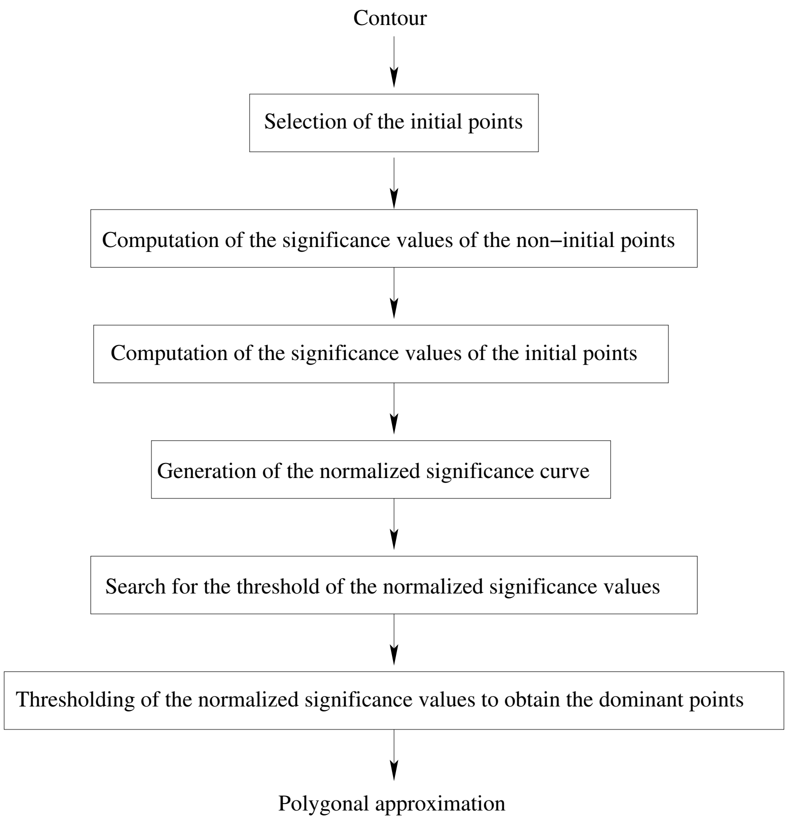
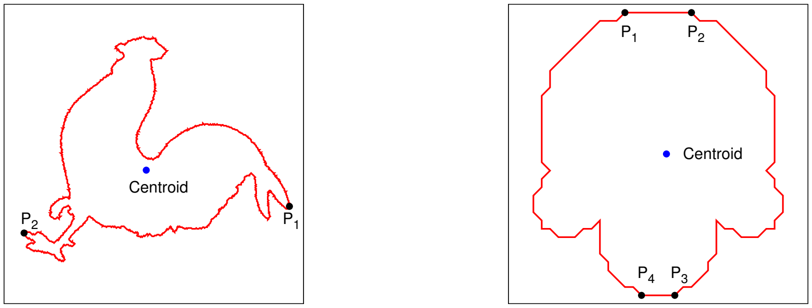

Contour
Selection of the initial points
Computation of the significance values of the non–initial points
Computation of the significance values of the initial points
Generation of the normalized significance curve
Search for the threshold of the normalized significance values
Thresholding of the normalized significance values to obtain the dominant points
Polygonal approximation

Figure 2: Framework of the proposed algorithm for polygonal approximation.



(a) *chicken-5*

(b) *semicircles*

Figure 3: *First step.- Selection of initial points for the contour*: (a) P_1 is the farthest point to the centroid and P_2 is the farthest point to P_1; (b) P_1 and P_2 are the farthest points to the centroid and P_3 and P_4 are the farthest points to the previous ones, respectively.

2. Run the recursive procedure: *compute_significance_values*

   - The following recursive procedure

     $$\textit{compute\_significance\_values}(P,P')$$

     is run for every two consecutive initial points P_i and $P_{(i+1)\ mod\ n}$ ($\forall i \in \{1, \ldots, n\}$):

     (a) Compute the maximum distance d from the contour points located between P and P' to the segment defined by P and P'.

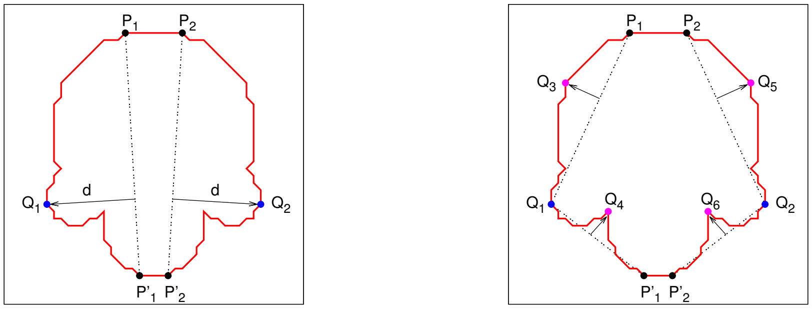
     (b) If d is equal to 0.0, then assign the significance value 0.0 to every contour point located between P and P'. In this case, the recursive process will stop.

     (c) On the contrary, if d is not equal to 0.0, then

       i. Select the farthest contour point(s) to the segment defined by P and P'.
         - In most cases, there will be only a single point, and the procedure will continue in the step ii.
         - However, there could be more than one point, for instance, if the contour has a straight side parallel to the segment defined by P and P'. In this situation, select from among them only the farthest points to the centroid of the contour, which will be considered as *candidate points* (see Figure 5).
         - This step takes into account the symmetry of the contour.

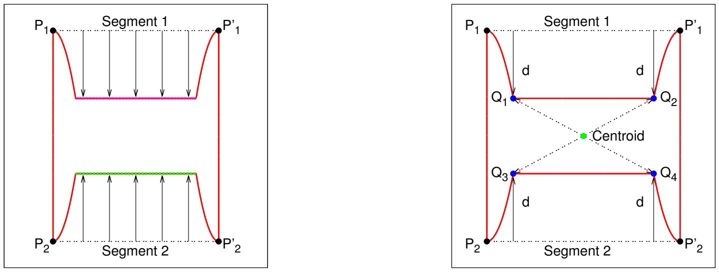
       ii. Be $Q_1, Q_2, \ldots, Q_m$ the selected *candidate points*:
         - Assign the maximum distance d as *significance value* of every point Q_i, $i \in \{1, \ldots, m\}$.
         - *Run* the recursive process *compute_significance_values* with every pair of points P and Q_1, Q_1 and Q_2, $\ldots$, Q_{m-1} and Q_m and, finally, Q_m and P'.

In summary, the significance value of every *non-initial* point is its deviation error d (see Figure 4) and takes into account the symmetry of the contour. The computational complexity of this step is $\mathcal{O}(nlog(n))$ in the best case, and $\mathcal{O}(n^2)$ in the worst case [9].



(a) (b)

Figure 4: *Second step.- Computation of the significance of the non-initial points*: performance of the recursive process *compute_significance_values* on the contour *semicircles*: (a) 2 first candidate points Q_1 and Q_2 and their significance value d; (b) 4 following candidate points: Q_3-Q_6.



(a) (b)

Figure 5: *Second step.- Computation of the significance of the non-initial points*: contour with straight sides (a) farthest points to the segments defined by the *initial points* P_1, P_1', P_2, P_2', (b) farthest points to the centroid among the previous farthest points to the segments: d is the significance value of the candidate points Q_1-Q_4.

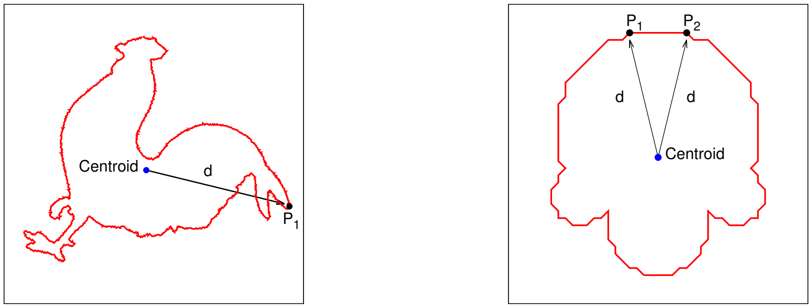
3.3. Third step: computation of the significance value of the initial points

The aim of this step is to assure that the initial points will have the greatest significance value. Initially, the *maximum* of the significance value of the *non-initial* points is computed: Sig_{max}.

- If Sig_{max} is equal to 0.0, then the significance value of the *initial* points will be 1.0. This situation occurs with artificial contours in which the initial points define an exact polygonal approximation.

- On the contrary, if Sig_{max} is not equal to 0.0, then the significance value of the *initial* points will be the maximum between Sig_{max} and d, where d is the maximum distance from the centroid to the contour points (see Figure 6).

The computational complexity of this step is $\mathcal{O}(n)$, because the maximum significance value of the contour points must be computed. The maximum distance of the contour points to the centroid was computed in the first step.



(a) *chicken-5*

(b) *semicircles*

Figure 6: *Third step.- Proposed method for the computation of the significance of the initial points*: d is the maximum distance from the centroid to the contour points.

3.4. Fourth step: the normalized significance curve

The *normalized significance curve* is the resulting plot of the polygonal approximation error, defined as a function of the number of dominant points (see Figure 7). This curve is strictly downward with a very pronounced or steep slope. At the beginning, the error reduction is significant; but, later, the error decreases when the small details of the contours are approximated.

The normalization of the significance curve attempts to equate the influence of the number of points and the values of significance. The normalization consists of three steps:

1. The contour points are sorted in descending order according to their significance values.
2. The significance values are normalized in the range $[0.0, 1.0]$ dividing by the maximum significance value.
3. The number of contour points are normalized in the range $[0.0, 1.0]$ dividing by the number N of the contour points.

This normalized significance curve is used in the fifth step (see Section 3.5) to locate the *corner point*, when the behavior of the curve changes from the approximation to the important parts of the contour to the approximation to the small details.

The computational complexity of this step is $\mathcal{O}(nlog(n))$, because the normalization is a linear process $\mathcal{O}(n)$ and the normalized significance value can be sorted using an algorithm with computational complexity $\mathcal{O}(nlog(n))$.

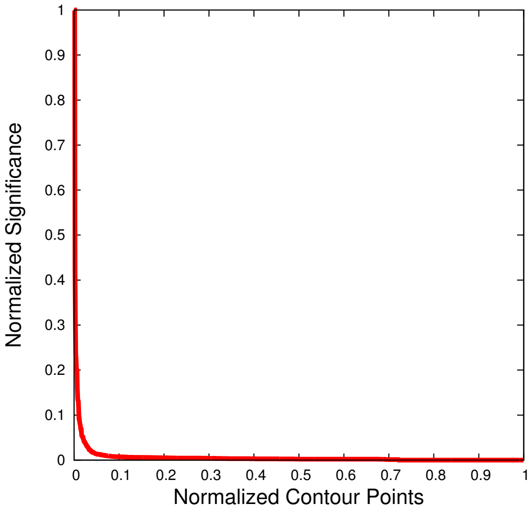


Figure 7: *Fourth step.- The normalized significance curve*: the plot shows the normalized significance curve generated by the contour *chicken-5* (N = 1364 points).

3.5. Fifth step: search for the threshold of the normalized significance values

The aim of this fifth step is to locate the *corner point* of the normalized significance curve. This *corner point* attempts to equate the influence of the number of points of the final polygonal approximation and its deviation error. The ordinate y of the *corner point* P is the chosen value h to threshold the normalized significance values of the contour points (see Section 3.6).

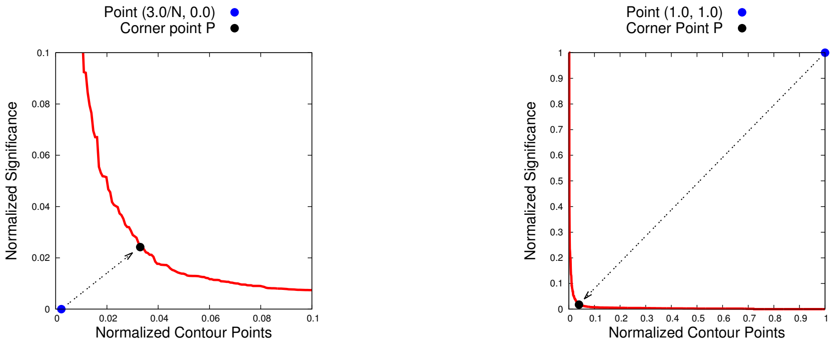
The *corner point* is searched within a *region of interest*, which is defined by the interval [*Left extreme point, Right extreme point*] of the abscissa axis, where the *Left extreme point* is the greatest normalized point with normalized significance value equal to 1.0, and the *Right extreme point* is the smallest normalized point with normalized value of significance equal to 0.0.

Four thresholding methods have been proposed for searching the *corner point*:

1. *Proximity* method: this method searches for the point $P(x, y)$ of the normalized significance curve with the *smallest* distance to the point[1] $(3.0/N, 0.0)$, where N is the number of contour points (see Figure 8(a)).
2. *Distance* method: this method searches for the point $P(x, y)$ of the normalized significance curve with the *greatest* distance to the point $(1.0, 1.0)$ (see Figure 8(b)).
3. *Rosin* method: this method has been inspired by the Rosin method for unimodal thresholding [33] and searches for the point $P(x, y)$ in the region of interest of the significance curve with the *greatest* distance to the straight line defined by the points *(left extreme, 1.0)* and *(right extreme, 0.0)*. This straight line is called *Rosin's line* (see Figure 9(a)).
4. *Adaptive* method: this method searches for the *farthest* point $P(x, y)$, in the region of interest of the significance curve, to the *Adaptive point* $Q(x_0, 1.0)$, where the project point $Q'(x_0, 0.0)$ is the first point in ascending order with normalized significance value equal to 0.0 (see Figure 9(b)).

In a few special cases, the *Adaptive* thresholding method does not work properly when the number of break points is much less than the number of contour points. In such a case, the distance from the *Adaptive point* $Q(x_0, 1.0)$ to the *farthest* point $P(x, y)$ is equal to 1.0 and greater than the distance from the *Adaptive point* to the *corner* point (see Figure 10(b)). The

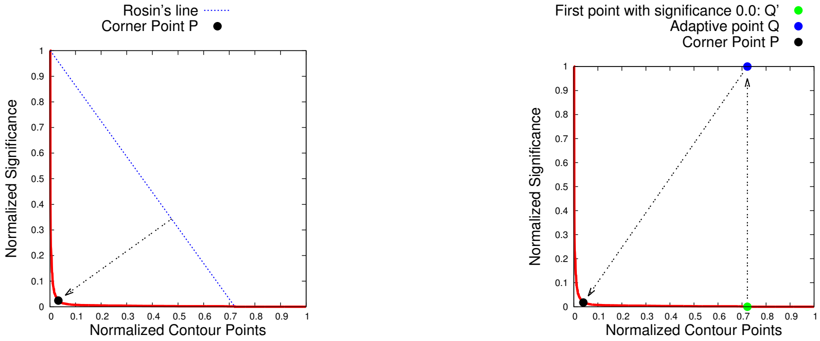
[1] The value of 3 has been chosen because the smallest polygon is the triangle (3 sides).



(a)

(b)

Figure 8: *Fifth step*: (a) *Proximity* thresholding method: the threshold h is the ordinate y of the nearest point P to the point (3.0/N, 0.0), where N is the number of contour points (zoom on $[0.0, 0.1]$); (b) *Distance* thresholding method: the threshold h is the ordinate y of the farthest point P to the point $(1.0, 1.0)$.



(a)

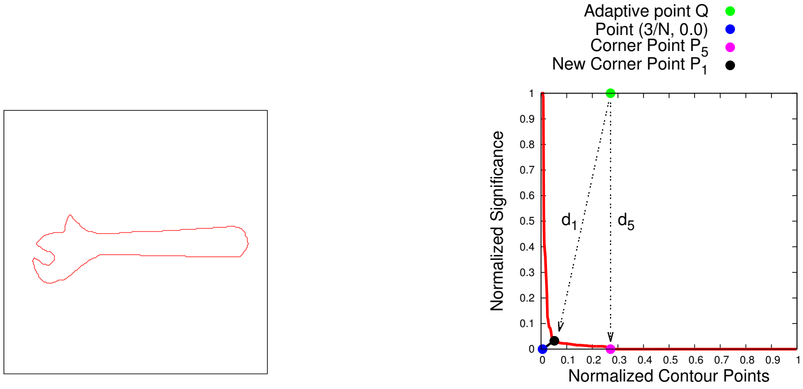
(b)

Figure 9: *Fifth step*: (a) *Rosin* thresholding method: the threshold h is the ordinate y of the farthest point P to the Rosin's line; (b) *Adaptive* thresholding method: the threshold h is the ordinate y of the farthest point P to the *Adaptive point* Q.

threshold h proposed by the *Adaptive* method is 0.0 and all the break points would be considered as dominant points, which would be undesirable. In other to avoid this drawback, if the threshold h proposed by the *Adaptive* method is 0.0, then the threshold h proposed by the *Proximity* method will be used instead.

The performances of these four thresholding methods have been analyzed empirically (see the first experiment in Section 4.2) and the *Adaptive* method, with the modification for special cases described previously, has obtained the best results and, therefore, it has been chosen for the fifth step of the new algorithm.

The computational complexity of this fifth step is linear: $\mathcal{O}(n)$.



(a)

(b)

Figure 10: Special case: (a) contour *adjustable-spanner* (N = 484 points; only 131 break points = 27,07 %) and (b) its normalized significance curve. The corner point P_4 proposed by the fourth method is changed by the corner point P_1 proposed by the first method. ($d_1 = 0.9919$, $d_4 = 1.0$)).

3.6. Sixth step: thresholding of the normalized significance values

In order to obtain the *dominant points*, a thresholding method is applied to the *normalized* significance values of the contour points. If $NSig(P)$ represents the normalized significance value of the point P and h is the threshold proposed by any of the thresholding methods (see Section 3.5) then the dominant points of the contour will be the points with normalized significance value greater than the threshold h:

$$Dominant\_Point(P) = \begin{cases} True & \textit{if } (NSig(P) > h) \\ \\ False & \textit{On the contrary.} \end{cases} \quad (1)$$

The computational complexity of this step is linear: $\mathcal{O}(n)$.

4. Experiments and results

Two experiments have been developed to measure the quality of the new algorithm for dominant points detection:

- In the first experiment (see Section 4.2), the performances of the four proposed thresholding methods (see Section 3.5) have been analyzed and the *Adaptive* thresholding method has obtained the best results. This method has been chosen for the fifth step of the new algorithm for polygonal approximations.

- In the second experiment (see Section 4.3), the new proposal, using the *Adaptive* thresholding method, has been compared with the non-parametric dominant point detection algorithms proposed by Prasad et al. [30].

The database of shapes "MPEG-7 Core Experiment CE-Shape-1 Test Set (Part B)" [12] has been used in the experiments [26, 27]. This database is available in [16] and contains 1400 images, classified into 70 categories, and each category includes 20 samples, with different rotation, size and position, and even image resolution [15]. The shapes contours have been extracted using the algorithm implemented in OpenCV 2.4.6. The contours obtained from the shapes of the database are affected by digitization and noise problems that occur in real situations.

4.1. Performance quality measures

In order to evaluate the quality of the polygonal approximations generated by the algorithms, several measures have been proposed [6, 14, 26, 27, 32]. *Compression ratio* $CR = \frac{n}{n_d}$, where n is the number of contour points and n_d is the number of dominant points, measures the normalized compression rate of the polygonal approximation. On the other hand, the *fitting* of the polygonal approximation to the original contour can be evaluated using several measures: (a) *Integral square error* $ISE = \sum_{i=1}^{n} e_i^2$, where e_i is the distance of contour point P_i to the polygonal approximation; (b) *Maximum error* $E_\infty = \max_{i \in \{1,\cdots,n\}}\{e_i\}$. ISE may not take into account some relevant information of the contour when hides large deviation at particular point due to closeness of approximating polygon at other parts of the curve. Such approximation behavior is not desirable. E_∞ is proposed to solve this lack of accuracy of ISE [21].

In order to combine the *number of vertices* and the *fitting* of the polygonal approximation, some measures have been proposed. Sarkar [36] proposed the *Figure of Merit* $FOM = \frac{CR}{ISE}$ to make the tradeoff between the compression ratio (CR) and the total distortion (ISE) caused [22]. Rosin [32] showed that the two terms CR and ISE used by FOM are not balanced, causing the measure to be biased towards approximations with lower ISE (which can be easily attained by increasing the number of detected dominant points). This drawback becomes more evident for real contours, which usually contain large numbers of points. Hence, FOM is not the best measure for comparing contours with different numbers of dominant points. The *weighted sum of squared error* $WE = \frac{ISE}{CR}$ is defined as the inverse of FOM [38, 39]. The *weighted maximum error* $WE_\infty = \frac{E_\infty}{CR}$ has also been proposed [18, 21, 39]. Technically, WE and WE_∞ are similar to FOM and suffer similar problems [21].

A parameterized version $WE_n = \frac{ISE}{(CR)^n}$ has also been proposed [5, 20, 25, 26] to balance the contribution of ISE and CR, where $n = 1, 2, 3$. Carmona [4] demonstrated that the value $n = 2$ obtains the best performance.

Rosin [32] proposed a new measure to avoid the drawback of FOM: *Merit* $= \sqrt{\mathit{Fidelity} \times \mathit{Efficiency}}$, where $\mathit{Fidelity} = \frac{E_{opt}}{E_{appr}} \times 100$ and $\mathit{Efficiency} = \frac{N_{opt}}{N_{appr}} \times 100$ where E_{appr} and N_{appr} are the error and the number of dominant points of suboptimal polygonal approximation, E_{opt} is the error produced by the optimal algorithm with the same number of dominant points and N_{opt} represents the number of dominant points that would require an optimal algorithm to produce the same error [28]. The *Fidelity* measures how well the polygon obtained by the algorithm to be tested fits the curve relative to the optimum polygon, in terms of the approximation error. The *Efficiency* measures the compactness of the polygon obtained by the algorithm to be tested, relative to the optimum polygon that incurs the same error [32].

The Rosin's evaluation can compare results of different algorithms with different number of dominant points. Nevertheless, Masood [22] pointed out that this method also suffers a few weaknesses. The polygon that consists of just *break points* will produce *Fidelity* = 100, *Efficiency* = 100 and *Merit* = 100. It means that the set of break points taken as dominant points will produce a perfect approximation, whereas this type of approximation is of no practical use since its compression ratio is very low. Carmona [6] proposed a new measure for assessing polygonal approximation of curves. This measure uses the method of Pérez and Vidal [28] and the optimization of an objective function based on the WE_2. The main drawback of these measures proposed by Rosin [32] and Carmona [6] is that they need optimal solutions which are computationally very expensive [28].

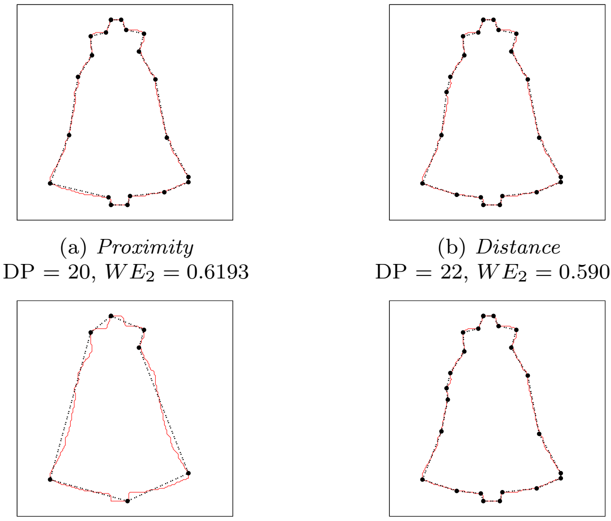
In summary, when the number of dominant points (DP) is the same, ISE and E_∞ are the most appropriate quantitative measures for comparison of polygonal approximations [21]. If the number of DP is different, then the parameterized WE_2 has showed better performance [4]. In the experiments, for the sake of completeness, the values of DP, CR, ISE, E_∞, WE, WE_2 and WE_3 have been used (see tables 1 and 2).

4.2. First experiment: selection of the best thresholding method of the fifth step of the new algorithm

The new algorithm for polygonal approximations consists of six steps (see section 3), but four *thresholding* methods have been proposed for the fifth step (see Section 3.5): *Proximity*, *Distance*, *Rosin* and *Adaptive* methods. The aim of this first experiment is to select the best thresholding method.

Taking into account the proposed thresholding methods, four versions of the new algorithm have been applied to 1400 contours extracted from the shapes of the MPEG-7 Core Experiment CE-Shape-1 Test Set (Part B) database [12]. Figures 11-13 show some polygonal approximations generated using the four thresholding methods. These figures show that the *Adaptive* thresholding method generates the best polygonal approximations. In order to facilitate the comparisons, the figures indicate the number of dominant points (DP) and the value of WE_2.

The quantitative evaluation of the thresholding methods is based on Table 1, that shows the average values of the quality measures. The *Adaptive* method has obtained the best results with the measures WE_∞, WE, and WE_2 and a very good result, in relation to the best, with the measure WE_3. The *Adaptive* method generates polygonal approximations with a few more dominant points (DP) than the other methods, but the fitting to the contour is significantly improved (see Figures 11-13). Therefore, the *Adaptive*



(a) *Proximity*
DP = 20, $WE_2 = 0.6193$

(b) *Distance*
DP = 22, $WE_2 = 0.5908$

(c) *Rosin*
DP = 7, $WE_2 = 0.6594$

(d) *Adaptive*
DP = 23, $WE_2 = 0.5334$

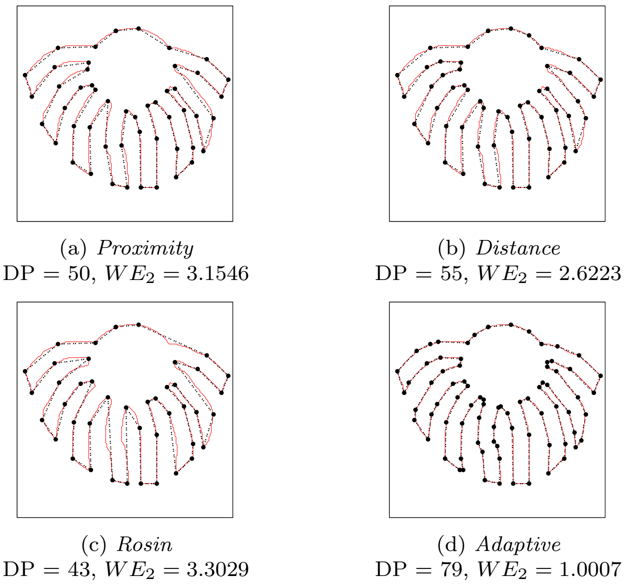
Figure 11: First experiment: polygonal approximations for *bell-7* (N = 406 points) using the proposed thresholding methods.

Table 1: First experiment: performance comparison of the new algorithm using different thresholding methods.

Thresholding method	Average values							
	DP	CR	E_∞	ISE	$WE\infty$	WE	WE_2	WE_3
Proximity	32.09	41.16	4.25	3908.34	0.122	91.812	2.428	0.073
Distance	33.18	40.97	3.98	3713.35	0.115	81.531	2.134	**0.066**
Rosin	25.01	56.03	6.22	9576.40	0.131	145.134	2.868	0.067
Adaptive	45.07	30.97	2.76	2024.67	**0.105**	**53.976**	**1.751**	0.069

Note: the average number of contour points of MPEG-7 CE-Shape-1 (Part B) is 1271.04 points.

method has been chosen as the *thresholding* method of the fifth step of the new algorithm. Once the best thresholding method has been chosen, the new algorithm can *automatically* generate polygonal approximations of closed digital planar curves without using any parameters.


(a) *Proximity*
DP = 50, $WE_2 = 3.1546$

(b) *Distance*
DP = 55, $WE_2 = 2.6223$

(c) *Rosin*
DP = 43, $WE_2 = 3.3029$

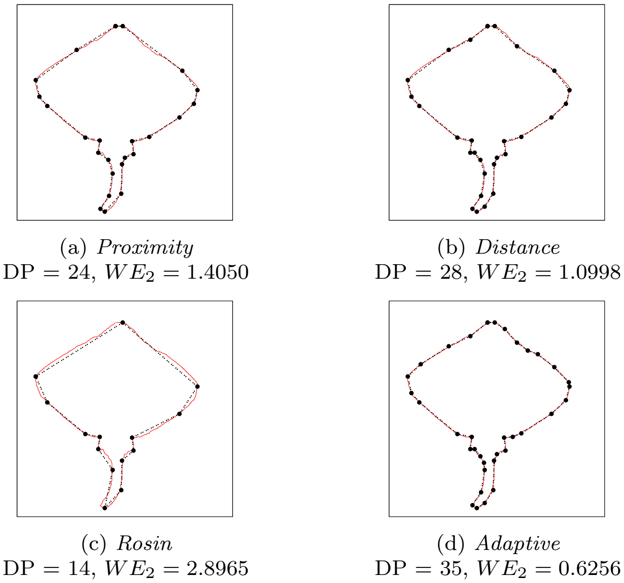
(d) *Adaptive*
DP = 79, $WE_2 = 1.0007$

Figure 12: First experiment: polygonal approximations for *octopus-14* (N = 1211 points) using the proposed thresholding methods.

4.3. Second experiment: comparison with the non-parametric algorithms proposed by Prasad

In the second experiment, the new proposal for polygonal approximations, using the *Adaptive* thresholding method, has been compared with the versions of RDP, Carmona and Masood algorithms proposed by Prasad et al. [30]. These algorithms have been selected because they can automatically generate polygonal approximations of digital planar curves.

The computational complexity of the proposed algorithm is $\mathcal{O}(nlog(n))$, which is similar to the computational complexity of the RDP method [9]. The computational complexity of the Carmona algorithm is $\mathcal{O}((m - n_d)n)$, where n is the number of the contour points, m is the number of initial break points, and n_d is the number of final dominant points [5]. The computational


(a) *Proximity*
DP = 24, $WE_2 = 1.4050$

(b) *Distance*
DP = 28, $WE_2 = 1.0998$

(c) *Rosin*
DP = 14, $WE_2 = 2.8965$

(d) *Adaptive*
DP = 35, $WE_2 = 0.6256$

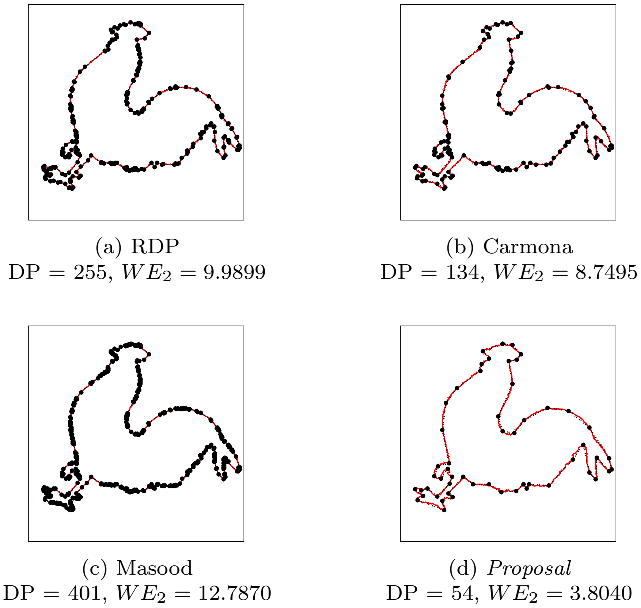
Figure 13: First experiment: polygonal approximations for *ray-17* (N = 689 points) using the proposed thresholding methods.

complexity of the Masood method can be given as $\mathcal{O}((m - n_d)n^2)$ owing to its optimization process [23].

The algorithms have been applied to 1400 contours extracted from the shapes of MPEG-7 Core Experiment CE-Shape-1 Test Set (Part B) database [12] to analyze their performances. The figures 14-18 show some polygonal approximations generated by these algorithms. Besides, the well-known contour *semicircles* has been used to show the performances of the algorithms with symmetric contours (see Figure 19). In order to facilitate the comparisons, the figures indicate the number of dominant points (DP), and the value of WE_2.

The figures 14(d)-18(d) show that the polygonal approximations generated by the new proposal properly fit to the contours. On the contrary, the performance of the algorithms proposed by Prasad et al. depends on the contours. For instance, these algorithms generate many dominant points for some contours (see Figures 14, 16 and 17) and very few dominant points for other contours (see Figures 17(c)) and 18(b) and (c)).

The new proposal is not affected by the digitization noise of the contours used in the experiments (see, for instance, Figure 15). In addition, the new



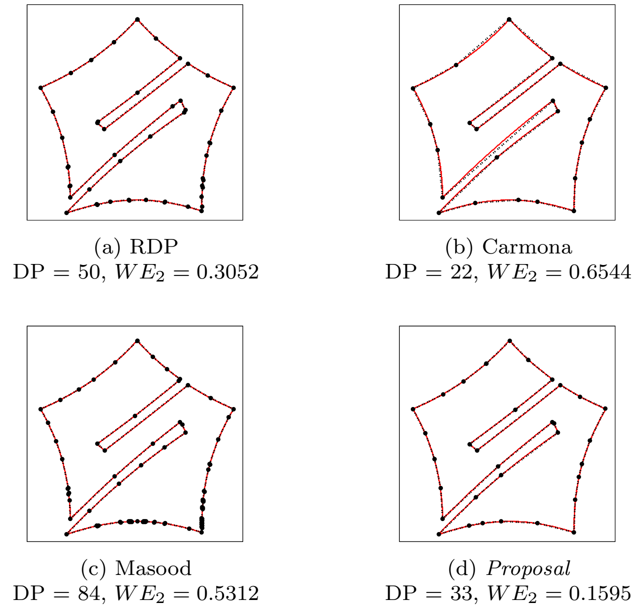
(a) RDP
DP = 255, $WE_2 = 9.9899$

(b) Carmona
DP = 134, $WE_2 = 8.7495$

(c) Masood
DP = 401, $WE_2 = 12.7870$

(d) *Proposal*
DP = 54, $WE_2 = 3.8040$

Figure 14: Second experiment: polygonal approximations for *chicken-5* (N = 1364 points).



(a) RDP
DP = 50, $WE_2 = 0.3052$

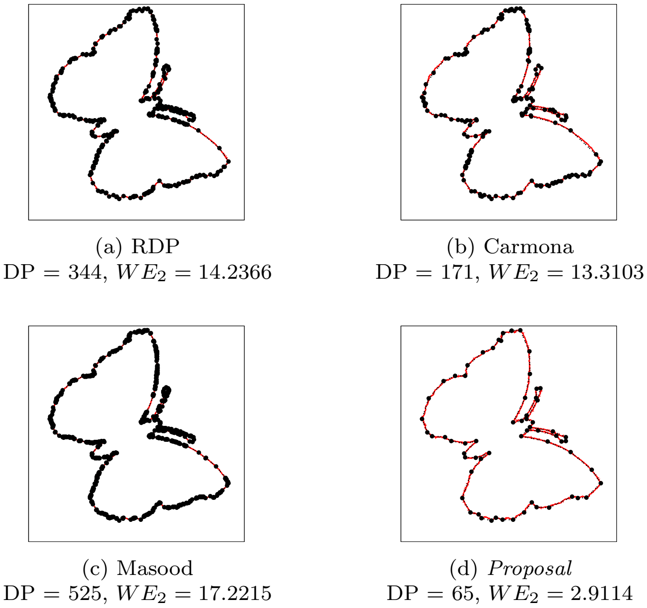
(b) Carmona
DP = 22, $WE_2 = 0.6544$

(c) Masood
DP = 84, $WE_2 = 0.5312$

(d) *Proposal*
DP = 33, $WE_2 = 0.1595$

Figure 15: Second experiment: polygonal approximations for *device6-9* (N = 1590 points).

algorithm takes into account the symmetry of the contours, whereas the versions proposed by Prasad et al. do not. See the polygonal approximations generated for the contour *semircicles* in the Figure 19.



(a) RDP
DP = 344, $WE_2 = 14.2366$

(b) Carmona
DP = 171, $WE_2 = 13.3103$

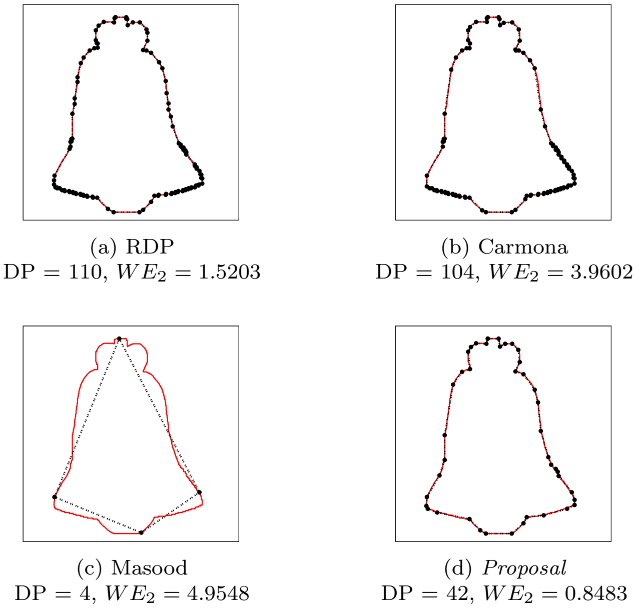
(c) Masood
DP = 525, $WE_2 = 17.2215$

(d) *Proposal*
DP = 65, $WE_2 = 2.9114$

Figure 16: Second experiment: polygonal approximations for *butterfly-13* (N = 1786 points).

The table 2 shows the average values of the quality measures obtained by the algorithms. The performance of the algorithms depends on the quality measures. The new proposal has obtained the best results for WE_2 and WE_3, and the second best results for WE_∞ and WE. This table also shows that the new method generates far fewer dominant points (DP) than the other methods. In our opinion, this experiment shows, once again, that WE (inverse of FOM) is not the best measure for comparing polygonal approximations with different numbers of dominant points because it is biased towards approximations with lower ISE, what it can be easily achieved by increasing the number of dominant points [20].

In summary, the second experiment has shown that the new proposal has good performance for generating polygonal approximations of 2D closed contours, and it has outperformed the non-parametric versions proposed by Prasad et al. [30].



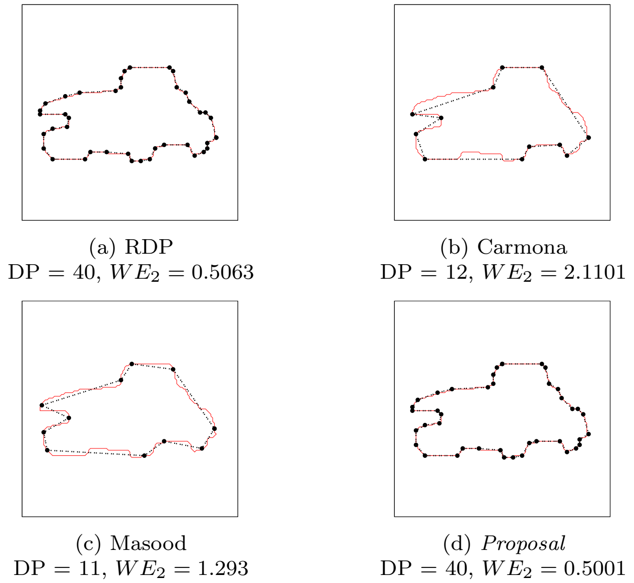
(a) RDP
DP = 110, $WE_2 = 1.5203$

(b) Carmona
DP = 104, $WE_2 = 3.9602$

(c) Masood
DP = 4, $WE_2 = 4.9548$

(d) *Proposal*
DP = 42, $WE_2 = 0.8483$

Figure 17: Second experiment: polygonal approximations for *bell-10* (N = 1202 points).



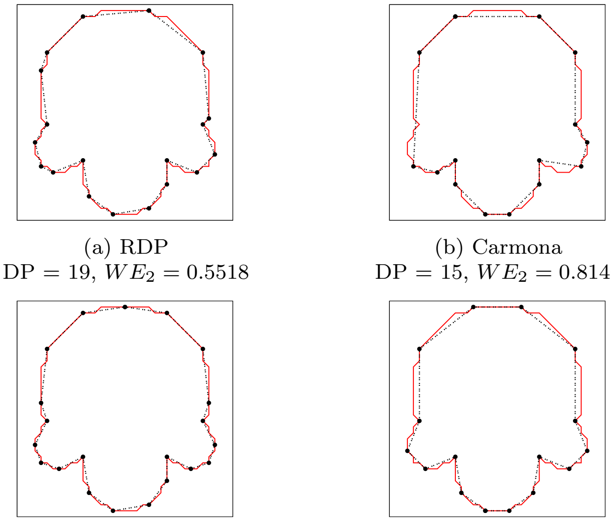
(a) RDP
DP = 40, $WE_2 = 0.5063$

(b) Carmona
DP = 12, $WE_2 = 2.1101$

(c) Masood
DP = 11, $WE_2 = 1.293$

(d) *Proposal*
DP = 40, $WE_2 = 0.5001$

Figure 18: Second experiment: polygonal approximations for *truck-07* (N = 277 points).



(a) RDP
DP = 19, $WE_2 = 0.5518$

(b) Carmona
DP = 15, $WE_2 = 0.8140$

(c) Masood
DP = 21, $WE_2 = 0.3941$

(d) Proposal
DP = 16, $WE_2 = 0.7711$

Figure 19: Second experiment: polygonal approximations for *semicircles* (N = 102 points). Only the new *proposal* generates a symmetric polygonal approximation.

Table 2: Second experiment: performance comparison of the algorithms.

Algorithm	Average values							
	DP	CR	E_∞	ISE	$WE\infty$	WE	WE_2	WE_3
RDP	110.12	17.78	1.23	219.34	**0.103**	**19.554**	2.266	0.313
Carmona	67.70	32.59	3.63	2168.16	0.159	61.959	3.164	0.242
Masood	132.15	39.52	5.10	91376.80	0.114	279.230	3.451	0.505
Proposal	45.07	30.97	2.77	2024.67	0.105	53.976	**1.751**	**0.069**

Note: the average number of contour points of MPEG-7 CE-Shape-1 (Part B) is 1271.04 points.

5. Conclusions

The present paper has proposed a new approach for polygonal approximations of closed digital planar curves. The new proposal consists of six steps and is focused on detecting dominant points which are usually used for automatically generating a polygonal approximation of the given closed curves. The significance values of the contour points are computed using a new version of the RDP method taking into account the *symmetry* of the contour, which is a common feature of the natural shapes. The final dominant points are located using a new *Adaptive* method to threshold the normalized significance values of the contour points. The proposed algorithm is *automatic* and *non-parametric*, and its computational complexity is $\mathcal{O}(nlog(n))$. The experiments have shown that the new algorithm can generate efficient and effective polygonal approximations for digital planar curves.

Finally, the proposed approach based on a *thresholding* method can be applied by any dominant points detection algorithm that assigns a significance values to the contour points.

References

[1] E.J. Aguilera-Aguilera , A. Carmona-Poyato, F.J. Madrid-Cuevas, R. Muoz-Salinas, Novel method to obtain the optimal polygonal approximation of digitalplanar curves based on Mixed Integer Programming. J. Vis. Commun. Image R. 30 (2015) 106–116. http://dx.doi.org/10.1016/j.jvcir.2015.03.007.

[2] E. Ataer-Cansizoglu, E. Bas, J. Kalpathy-Cramer, G. C. Sharp, D. Erdogmus, Contour-based shape representation using principal curves, Pattern Recognition 46 (2013) 1140–1150. http://dx.doi.org/10.1016/j.patcog.2012.10.014.

[3] F. Attneave, Some informational aspects of visual perception, Phychol. Rev. 61 (1954) 189–193.

[4] A. Carmona-Poyato, N.L. Fernandez-Garcia, R. Medina-Carnicer, F.J. Madrid-Cuevas, Dominant point detection: a new proposal, Image and Vision Computing 23 (2005) 1226–1236. http://dx.doi.org/10.1016/j.imavis.2005.07.025.

[5] A. Carmona-Poyato, F.J. Madrid-Cuevas, R. Medina-Carnicer and R. Muñoz-Salinas, Polygonal approximation of digital planar curves through break point suppression, Pattern Recognition 43 (2010) 14–25. http://dx.doi.org/10.1016/j.patcog.2009.06.010.

[6] A. Carmona-Poyato, R. Medina-Carnicer, F. J. Madrid-Cuevas, R. Muñoz-Salinas, R, N. L. Fernández-García, A new measurement for assesing polygonal approximation of curvers, Pattern Recognition. 4 (2011) 44–54. http://dx.doi.org/10.1016/j.patcog.2010.07.029.

[7] D. H. Douglas, T. K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Cartographica: The International Journal for Geographic Information and Geovisualization 10 (2) (1973) 112–122. http://dx.doi.org/10.3138/FM57-6770-U75U-7727.

[8] K. Ebisch, A correction to the DouglasPeucker line generalization algorithm, Computers & Geosciences 28 (2002) 995–997. http://dx.doi.org/10.1016/S0098-3004(02)00009-2.

[9] J. Hershberger, J. Snoeyink, Speeding up the Douglas-Peucker line simplification algorithm, Proceedings of the Fifth International Symposium on Spatial Data Handling, P. Bresnahan (Ed.), Charleston SC, (1992) 134–143.

[10] J.-H. Horng, J.T. Li, An automatic and efficient dynamic programming algorithm for polygonal approximation of digital curves, Pattern Recognition Letters 23 (2002) 171–182. http://dx.doi.org/10.1016/S0167-8655(01)00098-8.

[11] J. M. Iñesta, M. Buendia, M.A. Sart, Reliable polygonal approximations of imaged real objects through dominant point detection, Pattern Recognition 31 (1998) 685–697. http://dx.doi.org/10.1109/34.709604

[12] S. Jeannin, M. Bober, Description of core experiments for MPEG-7 motion/shape,MPEG-7, ISO/IEC/JTC1/SC29/WG11/MPEG99/N2690, SeoulMarch 1999.

[13] A. Kolesnikov, P. Fränti, Polygonal approximation of closed discrete curves, Pattern Recognition 40 (2007) 1282 – 1293. http://dx.doi.org/10.1016/j.patcog.2006.09.002.

[14] A. Kolesnikov, t. Kauranne, Unsupervised segmentation and approximation of digital curves with rate-distortion curve modeling, Pattern Recognition 47(2014) 623–633. http://dx.doi.org/10.1016/j.patcog.2013.09.002.

[15] L.J. Latecki, R. Lakämper, and Ulrich Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 424–429, 2000.

[16] MPEG-7 Core Experiment CE-Shape-1 Test Set (Part B), Benchmarking image database for shape recognition techniques. http://www.cis.temple.edu/˜latecki/TestData/mpeg7shapeB.tar.gz

[17] S. Loncaric, A survey of shape analysis techniques, Pattern Recongnition 31(1998) 983–1001. http://dx.doi.org/10.1016/S0031-2023(97)00122-2.

[18] D. G. Lowe, Three Dimensional Object Recognition from Single Two Dimensional Images, Artificial Intelligence 31 (1987) 355–395. http://dx.doi.org/10.1016/0004-3702(87)90070-1.

[19] M. Marji, P. Siy, A new algorithm for dominant point detection and polygonization of digital curves, Pattern Recognition 36 (2003) 2239–2251. http://dx.doi.org/10.1016/S0031-3203(03)00119-5.

[20] M. Marji, P. Siy, Polygonal representation of digital planar curves through dominant point detection–a nonparametric algorithm, Pattern Recognition 37 (2004) 2113–2130. http://dx.doi.org/10.1016/j.patcog.2004.03.004.

[21] A. Masood, S. A. Haq, A novel approach to polygonal approximation of digital curves, Journal of Visual Communication and Image Representation 18 (2007) 264–274. http://dx.doi.org/10.1016/j.jvcir.2006.12.002.

[22] A. Masood. Dominant point detection by reverse polygonization of digital curves, Image and Vision Computing 26 (2008) 702–715. http://dx.doi.org/10.1016/j.imavis.2007.08.006.

[23] A. Masood, Optimized polygonal approximation by dominant point deletion, Pattern Recognition 41 (2008) 227–239. http://dx.doi.org/10.1016/j.patcog.2007.05.021.

[24] A. Melkman, J. O'Rourke, On polygonal chain approximation, in: G.T. Toussaint (Ed.), Computational Morphology, North-Holland, Amsterdam, 1998, pp. 87–95.

[25] T.P. Nguyen, I. Debled-Rennesson, A discrete geometry approach for dominant point detection, Pattern Recognition 44 (1) (2011) 32–44. http://doi:10.1016/j.patcog.2010.06.022

[26] M. T. Parvez, S. A. Mahmoud Polygonal approximation of digital planar curves through adaptive optimizations Pattern Recognition Letters 31 (2010) 1997–2005. http://dx.doi.org/10.1016/j.patrec.2010.06.007.

[27] M. T. Parvez. Optimized polygonal approximations through vertex relocations in contour neighborhoods. Image and Vision Computing 34 (2015) 1–10. http://dx.doi.org/10.1016/j.imavis.2014.10.012

[28] J. C. Perez, E. Vidal, Optimum polygonal approximation of digitized curves, Pattern Recognition Letters 15 (1994) 743–750. http://dx.doi.org/10.1016/0167-8655(94)90002-7.

[29] A. Pikaz, I. Dinstein, Optimal polygonal approximation of digital curves, Pattern Recognition 28 (1995) 373–379. http://dx.doi.org/10.1016/0031-3203(94)00108-X.

[30] D. K. Prasad, M. K. H. Leung, C. Quek, S.-Y. Cho, A novel framework for making dominant point detection methods non-parametric Image and Vision Computing 30 (2012) 843–859. http://dx.doi.org/10.1016/j.imavis.2012.06.010.

[31] U. Ramer, An iterative procedure for the polygonal approximation of plane curves, Comput. Graphics Image Process. 1 (3) (1972) 244–256. http://dx.doi.org/10.1016/S0146-664X(72)80017-0.

[32] P. L. Rosin, Techniques for assessing polygonal approximations of curves, IEEE Trans. Pattern Analysis and Machine Intelligence 19 (1997) 659–666. http://dx.doi.org/10.1109/34.601253.

[33] P. L. Rosin, Unimodal thresholding, Pattern Recognition 34 (2001) 2083–2096. http://dx.doi.org/10.1.1.159.9940

[34] M. Salotti, An efficient algorithm for the optimal polygonal approximation of digitized curves, Pattern Recognition Letters 22 (2001) 215–221. http://dx.doi.org/10.1016/S0167-8655(00)00088-X.

[35] M. Salotti, Optimal polygonal approximation of digitized curves using the sum of square deviations criterion, Pattern Recognition 35 (2002) 435–443. http://dx.doi.org/10.1016/S0031-3203(01)00051-6.

[36] D. Sarkar, A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves, Pattern Recognition 14 (1993) 959–964. http://dx.doi.org/10.1016/0167-8655(93)90004-W.

[37] C. Urdiales, A. Bandera, F. Sandoval, Non-parametric planar shape representation based on adaptive curvature functions, Pattern Recognition 35 (2002) 43–53. http://dx.doi.org/10.1016/S0031-3203(01)00041-3.

[38] W. Y. Wu, Dominant point detection using bending value, Image and Vision Computing 21 (2003) 517–525. http://dx.doi.org/10.1016/S0262-8856(03)00031-3.

[39] W. Y. Wu, An adaptive method for detecting dominant points, Pattern Recognition 36 (2003) 2231–2237. http://dx.doi.org/10.1016/S0031-3203(03)00087-6.

[40] D. Zhang, G. Lu, Review of shape representation and description techniques Pattern Recognition 37 (2004) 1 – 19. http://dx.doi.org/10.1016/j.patcog.2003.07.008.