An improved method for obtaining optimal polygonal approximations

A. Carmona-Poyato; F. J. Madrid-Cuevas; N. L. Fernández-García; R. Medina-Carnicer

Abstract— In computer vision, polygonal approximations of digital planar curves are important for a great number of applications. Given N ordered points on a Euclidean plane, an optimal method to obtain M points that defines a polygonal approximation with the minimum distortion is proposed. Some of the state-of-the-art methods solve optimally this problem, however a high computational burden is required.

In order to reduce the computational time of the state-of-art-methods, a new improved method for obtaining optimal polygonal approximations in closed curves is proposed. The authors present an improved version of a method proposed by them in 2017. The original method is an iterative method that uses the improved Salotti method for obtaining many local optimal polygonal approximations with a prefixed starting point, where each iteration improves to the previous iteration. Thus, the original method obtains the global optimal polygonal approximation by using the number of iterations required to reach the global optimum. In the new proposal, only three iterations by using the improved Salotti method are required.

Tests have shown that the global optimal polygonal approximation is obtained in more than 98% of cases and the computation time is significantly reduced compared with the original method.

Keywords— closed digital planar curve, optimal polygonal approximation, improved Salotti method, Pikaz method.

I. Introduction

Polygonal approximations of digital planar curves is an important problem in image processing, pattern recognition and computer graphics. They are used in variety applications like object recognition [1] and geographical information systems [2]. The main goal is to reduce the representation of the original curve decreasing the memory requirements and preserving important information about the curve.

The polygonal approximation problem can be formulated as two different optimization problems:

- min-#: Minimize the number of line segments M that forms a polygonal approximation, such that, the distortion error does not excess a threshold Ɛ.
- min-Ɛ: Minimize the distortion error for a polygonal approximation given the number of points M of the approximation.

In this paper a new method to optimally solve the min-Ɛ problem is proposed.

Usually, the error is measured using the value of the integral squared error (*ISE*). The value of *ISE* for a polygonal approximation is defined as

$$ISE = \sum_{i=1}^{N} e_i \tag{1}$$

where N is the number of points of the contour and e_i is the distance from P_i to the approximated line segment.

It is possible to calculate a polygonal approximation of a contour using two different kind of methods: optimization methods, which obtain optimal polygonal approximation, taking into account an optimization criterion; and heuristic methods, which obtain suboptimal polygonal approximation using some reasonable geometric or perceptual feature of the contour. The heuristic methods can be also divided in two categories: parametric methods, which use one or more parameters that need to be tuned; and non-parametric methods, which generate the polygonal approximations without using any parameter.

The optimization methods provide optimal solution; however, their computational complexities are higher than the computational complexities of the heuristic methods.

The present work proposes a new approach to obtain optimal polygonal approximations. This approach is based in an iterative method proposed by Carmona et al [3].

This work is arranged as follows. The section II describes the related works. The section III explains the new proposal. The experiments and results are described in section IV. Finally, the main conclusions are summarized in section V.

II. Related Works

In this section the methods used in our proposal will be described.

Pikaz et al. [4] proposed a suboptimal method based on a greedy iterative algorithm. First, all the collinear points of the original contour are deleted. The remaining points are called

This work has been developed with the support of the Research Project called TIN2016-75279 financed by Economy, Industry and Competitiveness Ministry of Spain.

A. Carmona-Poyato is with Department of Science Computer of Cordoba University. Spain (corresponding author. e-mail: ma1capoa@uco.es).

F. J. Madrid-Cuevas is with Department of Science Computer of Cordoba University. Spain (e-mail: ma1macuf@uco.es).

N. L. Fernández-García is with Department of Science Computer of Cordoba University. Spain (e-mail: ma1fegan@uco.es).

R. Medina-Carnicer is with Department of Science Computer of Cordoba University. Spain (e-mail: rmedina@uco.es).

breakpoints. This is the minimum number of points that produces an error equal to 0. The set of breakpoints is considered as the initial polygonal approximation. Then, by using an iterative procedure, many breakpoints are deleted until only *M* breakpoints remain. In each iteration, the breakpoint that produces the smallest error value is deleted. When a breakpoint is deleted the error associated with its two neighbors changes. This method is very fast *O(nlog n)* and it can produce polygonal approximation with any preset number of final points.

Masood [5] improved Pikaz method using a local optimization process search the optimal position of the remaining points that minimizes the distortion. This method is $O(N^2)$, however it does not guarantee that the solution obtained is optimal

Some optimization methods have been proposed [6], [7], [8], [9]. The proposals [6], [7], [8] solve this problem in $O(N^2)$-$O(MN^2)$ for open curves or closed curves when a starting point is prefixed. In the case of closed curves, the cited methods obtain a solution that depends on the starting point and only a local optimum for this prefixed starting point is obtained. For this reason, all the points of the contour should be tested as starting point for obtaining the global optimal polygonal approximation. Thus, the computational complexity increases one level and becomes $O(N^3)$-$O(MN^3)$.

Perez et al. [6] proposed an optimization method, based on dynamic programming, to solve the *min-ε* problem. Dynamic programming is an optimization method which selects a decision based on all possible previous states with a proper recurrence relation.

Perez et al. used the next recursive function to solve the problem:

$$E(N,M) = \min_{i=M-1,N-1}\left\{E(i,M-1)+e(P_i,P_N)\right\} \quad (2)$$

E(N, M) is the minimum error of approximate the first *N* points by using M points.

E(i, M-1) is the minimum error of approximate the first *i* points by using *M-1* points.

$e(P_i, P_j)$ is the error of approximate the curve segment between P_i and P_N by a straight line.

The computational complexity of this method is $O(MN^2)$ in open planar curves or closed curves when the starting point is prefixed. Its main drawback occurs in closed curves, when all the points should be considered as starting point, and its computational complexity is $O(MN^3)$.

Salotti [7] proposed a method based on the search of the shortest path in a graph using *A*-algorithm*. To reduce the search, Salotti proposed two procedures of pruning:

- To obtain a first rough polygonal approximation to estimate the value of a threshold on the maximum global error. Thus, nodes which cannot lead to optimal solutions are pruned. This rough polygonal approximation is obtained by using a suboptimal method with low computational complexity.
- To stop the exploration of successors of the shortest path in the graph as soon as possible. For this goal, Salotti proposed a simple solution stopping the exploration using a lower-bound.

This lower-bound is calculated using the linear regressions y/x and x/y to estimate least-square errors. So, he obtains the next expression for the lower-bound:

$$E_{low}^{P_i,P_j} = \frac{1}{2}\min\left\{E_{reg1}^{P_i,P_j}, E_{reg2}^{P_i,P_j}\right\} \quad (3)$$

where E_{reg1} and E_{reg2} are the errors calculated using the linear regressions y/x and x/y.

Horng et al [8] proposed a method to determine the initial point of the polygonal approximation. This heuristic method needs two iterations of the Dynamic Programming algorithm to construct a polygonal approximation. The main drawback is that algorithm assure the optimal solution with a low probability.

Carmona et al [9] proposed an improved version of Salotti method, by reducing the lower-bound using orthogonal least-square errors. This improvement reduces the computational time by 16%.

Using these improvements, the computational complexity is $O(N^2)$. Due to in this method the starting point is prefixed, and a local optimal polygonal approximation is obtained, all the points should be considered as starting point to obtain the global optimal solution. In this case the computational complexity is close to $O(N^3)$.

Aguilera et al [10] proposed a method to compute optimal polygonal approximations of closed curves based on Mixed Integer Programming. This method does not need the initial point of the polygonal approximation to compute the optimal polygonal approximation and the computation time does not depends on the number of points M of the polygonal approximation. Although this method obtains optimal polygonal approximations, its main drawback is that its computational complexity is NP.

To avoid these drawbacks, Carmona et al. [3] proposed an iterative method for obtaining optimal polygonal approximations with M points.

In the first iteration, two methods were used: a heuristic method [4] to obtain a suboptimal polygonal approximation with M points; and an optimal method [7] to obtain an optimal polygonal approximation with M points with a prefixed starting point. In this case, the value of the error of this approximation is used as value of pruning of the second iteration.

Moreover, the point of the obtained polygonal approximation that generates the greatest error if were eliminated, is used as starting point for the second iteration.

In the second iteration, an improved version of Salotti method was used for obtaining a local optimal polygonal approximation. In this iteration the error and the starting point obtained in the first iteration are used.

In the third and next iterations, the error obtained in the previous iteration is used as value of pruning. Further, five different procedures to select the starting point in each iteration was tested:

- By using the second point of the previous polygonal approximation as starting point.
- By using the third point of the previous polygonal approximation as starting point.

- By using the fourth point of the previous polygonal approximation as starting point.
- By using the *M/2-th* point of the previous polygonal approximation as starting point.
- By using the $M/2^{i-1}$*-th* point of the previous polygonal approximation as starting point in the *i-th* iteration, like the binary search.

In this work, it was shown that each iteration gets an error less than or equal to the error of the previous iteration and the optimal approximation is obtained in a few iterations. The best results were obtained using the fourth procedure to select the starting point.

For polygonal approximations with *M>15*, when Perez method was used in the first iteration, only two iterations were required to obtain the global optimal approximation in the 95% of the cases. When Pikaz method was used in the first iteration, six iterations were required to obtain the global optimal polygonal approximation in the 95% of the cases.

For polygonal approximations with *M<=15*, when Perez method was used in the first iteration, four iterations were required to obtain the global optimal approximation in the 95% of the cases. When Pikaz method was used in the first iteration, eight iterations were required to obtain the global optimal polygonal approximation in the 95% of the cases.

In conclusion, the best results were obtained using Perez method in the first iteration and the fourth procedure to select the starting point. These results will be compared with the new proposal.

III. Proposed Method

In this work, an improvement of [3] is proposed. To explain the new method, a polygonal approximation with M=8 of a synthetic contour will be used. The new approach uses an iterative method and it can be described as follows:

In the first iteration, Pikaz method is applied, thus a suboptimal polygonal approximation is obtained. The error of this approximation is used as value of pruning of the second iteration. Moreover, the point of the obtained polygonal approximation that would generate the greatest error if it is eliminated, is used as starting point for the second iteration. The choice of this point is due to the fact that this point is more likely to belong to the optimal approximation than the rest of the points of the approximation obtained in this iteration. Fig. 1 shows the eight points obtained. None of these points is optimal. In this case Point 2 will be used as starting

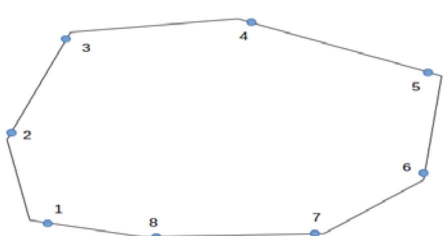

Figure 1. Approximation obtained after the first iteration.

point for the second iteration.

In the second iteration a second polygonal approximation is obtained. The error of the first iteration as value of pruning and the best point possible as starting point are used to apply the improved Salotti method. In this case, the obtained approximation is the best approximation for the selected starting point. Moreover, when a value of pruning is used, the computational time is highly reduced. Fig. 2 shows the eight points obtained. Points 4 and 5 are optimal. Point 5 will be used

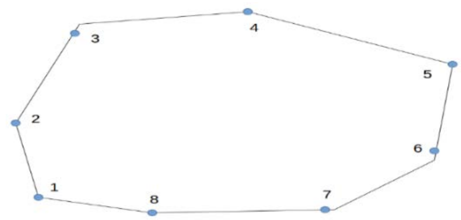

Figure 2. Approximation obtained after the second iteration.

as starting point for the third polygonal approximation.

In the third iteration, the improved Salotti method is also used. The error of the second iteration is used as value of pruning and the *M/2-th* point belong to the second approximation is used as starting point for this iteration. The choice of this point is because this point would be the least affected by a bad choice of the starting point of the second iteration. Moreover, Carmona et al [1] showed that this point produces the best results. Fig. 3 shows the eight points obtained. Points 4, 5, 8, 1, 2 are optimal. Point 5 will be used as starting point in polygonal approximation

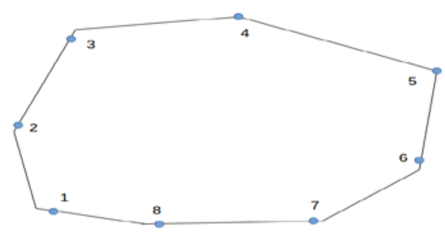

Figure 3. Approximation obtained after the third iteration.

In the fourth iteration, the improved Salotti method is only used in some of the points neighboring of the starting point of the previous iteration. In this way, a bad choice of the starting point in the third iteration is corrected. This iteration can be explained as follows:

- A neighborhood value *n* is used.
- *If (5 <= M <= 6), n = 2.*
- *If (6 < M <= 8), n = 4.*
- *If (8 < M <= 13), n= 6.*
- *If (M > 13), n = 11.*
- Let P_s be the starting point of the previous iteration.

  Improved Salotti method is applied between the point $P_l = P_s - n/2 - 1$ and the point $P_r = P_s + n/2 + 1$, corresponding to the polygonal approximation obtained in the third iteration. In this case, the error between P_l and

P_r, corresponding to the third iteration, is used as value of pruning. Due to the improved Salotti method is only applied to a reduced number of points, the computational time is highly decreased.

Fig. 4 shows the eight points obtained. All points are optimal. In this case the improved Salotti method has been applied between the points 3 and 7.

IV. Experimental Results and Discussion

The experiments have been carried out in a generic computer with processor Intel(R) Core(TM) i7-5500U CPU @ 2.4 GHz x 4, 7.7 GB of RAM memory. The proposal has been tested using 24 digital contours used in other works by the authors.

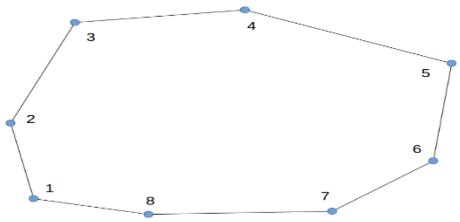

Figure 4. Approximation obtained after the fourth iteration

The number of points of these contours range from 554 to 2041. All the polygonal approximations between 5 and 50 points have been calculated. Due to this, 5520 polygonal approximations have been obtained. Fig. 5 and Fig. 6 show the digital contours used in this work.

To assess the proposed method and considering that the related methods have worse performance when the number of points of the polygonal approximation is small, polygonal approximations with few points (between 5 and 15) have been analyzed separately. Polygonal approximations between 16 and 50 points, usually produce balanced approximations like the original contours, with appropriate number of points (they are neither too high nor too low).

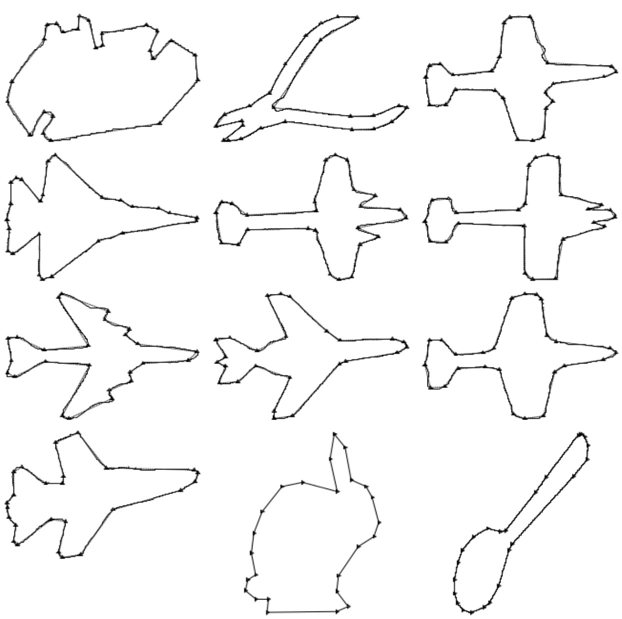

Figure 5. Some of the used contours and their optimal polygonal approximations for M=25.

To test the accuracy of the proposed method, the global optimal polygonal approximation has been obtained using improved Salotti method for all possible starting points.

For polygonal approximations with 5 <= M <= 15, the results have shown that the new proposal obtains the optimal polygonal approximation in 92.4% of cases while the original proposal obtains the optimal polygonal approximation in 85.9% of cases, when the same number of iterations is used. Moreover, the computational time is decreased by 41.23% in

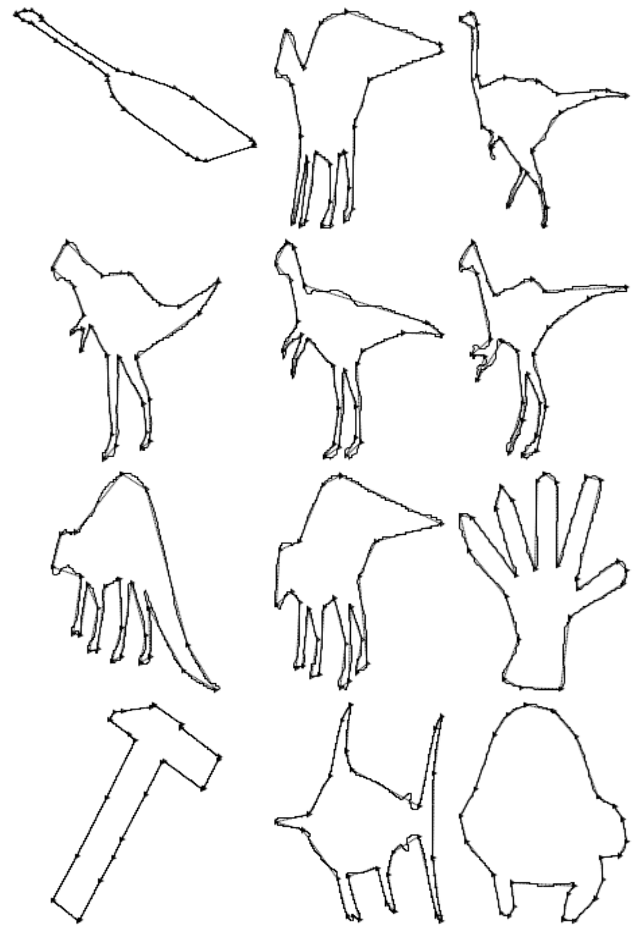

Figure 6. Some of the used contours and their optimal polygonal approximations for M=25.

the new proposal.
Figure 5 and 6 show some of the digital contours used in this work and their polygonal approximations for M= 30.

For polygonal approximations with 15 < M <= 50, the results have shown that the new proposal obtains the optimal polygonal approximation in 98.81% of cases while the original proposal obtains the optimal polygonal approximation in 97.5% of cases, when the same number of iterations is used. Moreover, the computational time is decreased by 34.9% in the new proposal.

To compare the computational time of the new proposal with the original proposal, the mean values of time for the new and original proposal have been calculated for each value of *M*, using the 24 contours. The obtained results are shown in Figure 7.

The computational time for the additional iteration in the new proposal is much lower than the computational time of the second and next iterations, because improved Salotti method is only applied to a small number of points of the original contour. For this reason, the computational time of the additional iteration has very little influence on the global computational time.

However, this additional iteration produces a reduction of the error of the polygonal approximation.

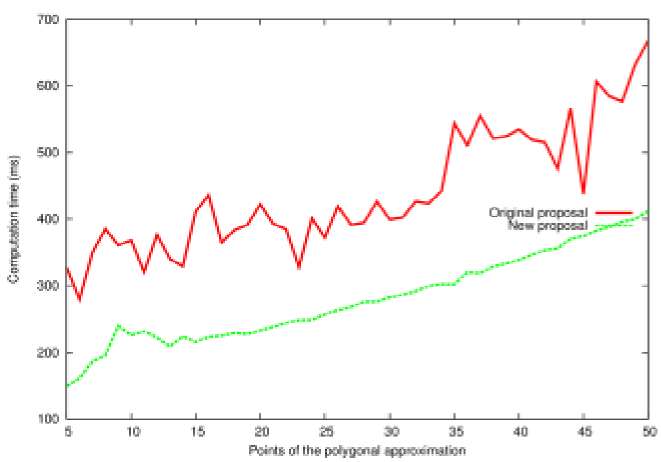

Figure 7. Computational time for the original proposal and the new proposal

V. Conclusions

The conclusions of this work can be summarized as follows:

A new improved method for obtaining optimal polygonal approximations in closed curves is proposed.
This new proposal is based in an old iterative method proposed by the authors in 2017.

The new proposal contains an additional iteration based in the improved Salotti method.

In this new iteration, this method is only applied to a reduced number of points of the original contour in order to reduce its computational time.

In the other hand, due to this method only use a reduced number of points as starting point, its computational time is much less than those methods that use all the points of the contour as starting point, and the optimal polygonal approximation is obtained with a high probability.

The new proposal is better than the method based on mixed integer programming, because the computational complexity of this method is NP.

Finally, the new proposal improves the original method considering both the computational time and the probability of obtaining the optimal solution:

- For polygonal approximations with 5 <= M <= 15, the results have shown that the new proposal obtains the optimal polygonal approximation in 92.4% of cases while the original proposal obtains the optimal polygonal approximation in 85.9% of cases and the computational time is decreased by 41.23% in the new proposal.
- For polygonal approximations with 15 < M <= 50, the results have shown that the new proposal obtains the optimal polygonal approximation in 98.81% of cases while the original proposal obtains the optimal polygonal approximation in 97.5% of cases and the computational time is decreased by 34.9% in the new proposal.

References

[1] O.M. Soysal, J. Chen, "Object recognition by spectral feature derived from canonical shape representation" Machine Vision and Applications, vol. 24, pp. 855–868, 2012.

[2] M. Chen, M. Xu, P. Franti, "A fast multiresolution polygonal approximation algorithm for GPS trajectory simplification" IEEE Transactions Image Processing, vol. 21, pp. 2770–2785, 2012.

[3] A. Carmona-Poyato, E. Aguilera-Aguilera, F.-J. Madrid-Cuevas, M. Marín-Jiménez and N. Fernández García, “New method for obtaining optimal polygonal approximations to solve the min-ε problem,” *Neural Computing and Applications*, vol. 28, pp. 2383–2394, 2017.

[4] A. Pikaz and I. Dinstein, “Optimal polygonal approximation of digital curves,” *Pattern Recognition*, vol. 28, pp. 371–379, 1995.

[5] A. Masood, "Dominant point detection by reverse polygonization of digital curves" Image and Vision Computing, vol. 26, pp. 702–715, 2008.

[6] J.-C. Perez and E. Vidal, “Optimum polygonal approximation of digitized curves,” *Pattern Recognition Letters*, vol. 15, pp. 743–750, 1994.

[7] M. Salotti, “Optimal polygonal approximation of digitized curves using the sum of square deviations criterion,” *Pattern Recognition*, vol. 35, pp. 435–443, 2002.

[8] J.-H. Horng and J.-T. Li, “An automatic and efficient dynamic programming algorithm for polygonal approximation of digital curves,” *Pattern Recognition Letters*, vol. 23, pp. 171–182, 2002.

[9] A. Carmona-Poyato, E. Aguilera-Aguilera, F.-J. Madrid-Cuevas, D. López-Fernández, “New method for obtaining optimal polygonal approximations,” *Iberian Conference on Pattern Recognition and Image Analysis*, pp. 149–156, 2015.

[10] E.J. Aguilera-Aguilera, A. Carmona-Poyato, F.J. Madrid-Cuevas, R. Muñoz Salinas, "Novel method to obtain the optimal polygonal approximation of digital planar curves based on mixed integer programming" Journal Visual Communication and Image Representation, vol. 30, pp.106–116, 2015.