UNIVERSITY OF CORDOBA

THESIS

Contribucciones sobre métodos óptimos y subóptimos de aproximaciones poligonales de curvas 2-D

A thesis submitted for the degree of Computer Science Doctor (PhD)

Author: D. Eusebio J. Aguilera Aguilera

Directors: Dr. Ángel Carmona Poyato
Dr. Francisco J. Madrid Cuevas

Córdoba, 2016

TITULO: *CONTRIBUCIONES SOBRE MÉTODOS ÓPTIMOS Y SUBÓPTIMOS DE APROXIMACIONES POLIGONALES DE CURVAS 2-D*

AUTOR: *Eusebio Jesús Aguilera Aguilera*

Campus de Rabanales
Ctra. Nacional IV, Km. 396 A
14071 Córdoba

www.uco.es/publicaciones
publicaciones@uco.es

TÍTULO DE LA TESIS: CONTRIBUCIONES SOBRE MÉTODOS ÓPTIMOS Y SUBÓPTIMOS DE APROXIMACIONES POLIGONALES DE CURVAS 2-D

DOCTORANDO/A: EUSEBIO JESÚS AGUILERA AGUILERA

INFORME RAZONADO DEL/DE LOS DIRECTOR/ES DE LA TESIS

(se hará mención a la evolución y desarrollo de la tesis, así como a trabajos y publicaciones derivados de la misma).

El doctorando ha cumplido los objetivos que se marcó en la realización de su tesis doctoral.

A lo largo de su realización ha realizado un estudio exhaustivo del estado del arte del campo de las aproximaciones poligonales y ha propuesto varios métodos novedosos, tanto óptimos como subóptimos, para obtener dichas aproximaciones a partir de una curva 2-D.

Dentro de las aportaciones realizadas sobre métodos subóptimos ha publicado un artículo en la revista indexada "Journal of Visual Communication and Image Representation", del segundo cuartil dentro de la categoría "Computer Science, Information Systems" y una comunicación en el congreso internacional "7th Iberian Conference on Pattern Recognition and Image Analysis".

Dentro de las aportaciones realizadas sobre métodos óptimos ha publicado un artículo en la revista indexada "Journal of Visual Communication and Image Representation" del segundo cuartil dentro de la categoría "Computer Science, Information Systems"; un artículo en la revista indexada "Neural Computing and Applications" del segundo cuartil dentro de la categoría "Computer Science, Artificial Intelligence"; un artículo en la revista indexada "Graphical models" del segundo cuartil dentro de la categoría "Computer Science, Software Engineering" y una comunicación en el congreso internacional "7th Iberian Conference on Pattern Recognition and Image Analysis".

Todas ellas han sido publicaciones novedosas y punteras dentro de la línea de investigación de la tesis realizada.

Por todo ello, se autoriza la presentación de la tesis doctoral.

Córdoba, 26 de FEBRERO de 2016

Firma del/de los director/es

Fdo.:ÁNGEL CARMONA POYATO Fdo.: FRANCISCO JOSÉ MADRID CUEVAS

Contents

List of Figures

Contribucciones sobre métodos óptimos y subóptimos de aproximaciones poligonales de curvas 2-D

Eusebio J. Aguilera Aguilera

Departamento de Informática y Análisis Numérico
Universidad de Córdoba

Resumen

Esta tesis versa sobre el análisis de la forma de objetos 2D. En visión artificial existen numerosos aspectos de los que se pueden extraer información. Uno de los más usados es la forma o el contorno de esos objetos. Esta característica visual de los objetos nos permite, mediante el procesamiento adecuado, extraer información de los objetos, analizar escenas, etc.

No obstante el contorno o silueta de los objetos contiene información redundante. Este exceso de datos que no aporta nuevo conocimiento debe ser eliminado, con el objeto de agilizar el procesamiento posterior o de minimizar el tamaño de la representación de ese contorno, para su almacenamiento o transmisión. Esta reducción de datos debe realizarse sin que se produzca una pérdida de información importante para representación del contorno original. Se puede obtener una versión reducida de un contorno eliminando puntos intermedios y uniendo los puntos restantes mediante segmentos. Esta representación reducida de un contorno se conoce como aproximación poligonal.

Estas aproximaciones poligonales de contornos representan, por tanto, una versión comprimida de la información original. El principal uso de las mimas es la reducción del volumen de información necesario para representar el contorno de un objeto. No obstante, en los últimos años estas aproximaciones han sido usadas para el reconocimiento de objetos. Para ello los algoritmos de aproximación poligonal se han usado directamente para la extracción de los vectores de características empleados en la fase de aprendizaje.

Las contribuciones realizadas por tanto en esta tesis se han centrado en diversos aspectos de las aproximaciones poligonales. En la primera contribución se han mejorado varios algoritmos de aproximaciones poligonales, mediante el uso de una fase de preprocesado que acelera estos algoritmos permitiendo incluso mejorar la calidad de las soluciones en un menor tiempo. En la segunda contribución se ha propuesto un nuevo algoritmo de aproximaciones poligonales que obtiene soluciones óptimas en un menor espacio de tiempo que el resto de métodos que aparecen en la literatura. En la tercera contribución se ha propuesto un algoritmo de aproximaciones que es capaz de obtener la solución óptima en pocas iteraciones en la mayor parte de los casos. Por último, se ha propuesto una versión mejorada del algoritmo óptimo para obtener aproximaciones

poligonales que soluciona otro problema de optimización alternativo.

Contribucciones sobre métodos óptimos y subóptimos de aproximaciones poligonales de curvas 2-D

Eusebio J. Aguilera Aguilera

Departamento de Informática y Análisis Numérico
University of Cordoba

Abstract

This thesis focus on the analysis of the shape of objects. In computer vision there are several sources from which we can extract information. One of the most important source of information is the shape or contour of objects. This visual characteristic can be used to extract information, analyze the scene, etc.

However, the contour of the objects contains redundant information. This redundant data does not add new information and therefore, must be deleted in order to minimize the processing burden and reducing the amount of data to represent that shape. This reduction of data should be done without losing important information to represent the original contour. A reduced version of a contour can be obtained by deleting some points of the contour and linking the remaining points by using line segments. This reduced version of a contour is known as polygonal approximation in the literature.

Therefore, these polygonal approximation represent a compressed version of the original information. The main use of polygonal approximations is to reduce the amount of information needed to represent the contour of an object. However, in recent years polygonal approximations have been used to recognize objects. For this purpose, the feature vectors have been extracted from the polygonal approximations.

The contributions proposed in this thesis have focused on several aspects of polygonal approximations. The first contribution has improved several algorithms to obtain polygonal approximations, by adding a new stage of preprocessing which boost the whole method. The quality of the solutions obtained has also been improved and the computation time reduced. The second contribution proposes a novel algorithm which obtains optimal polygonal approximations in a shorter time than the optimal methods found in the literature. The third contribution proposes a new method which may obtain the optimal solution after few iterations in most cases. Finally, an improved version of the optimal polygonal approximation algorithm has been proposed to solve an alternative optimization problem.

Agradecimientos

Quiero agradecer, en primer lugar, a mis directores de tesis, los doctores Ángel Carmona Poyato y Francisco José Madrid Cuevas. Todo el tiempo que me dedicaron y su esfuerzo que, sin duda, me han ayudado enormemente a finalizar este trabajo.

Asimismo, me gustaría dar las gracias a todos los miembros del grupo de investigación Aplicaciones de la Visión Artificial (AVA), que siempre me ayudaron a mejorar y dieron buenos consejos.

Agradezco a toda mi familia y amigos que siempre están ahí, en los buenos y en los malos momentos, para darme apoyo y cariño sin el que me habría sido imposible realizar este trabajo, y por supuesto sin los que no podría vivir.

Part I

Introduction

Chapter 1

Introduction

The improvements of the computers and other devices obtained in recent years has made computer vision an important research area. We can find several devices like smartphones, personal computers, etc, which incorporate a computer vision system. These systems use computer vision algorithms and methods to improve the human-computer interface. Moreover, other applications of computer vision are found in other fields of engineering, agriculture, medicine, etc.

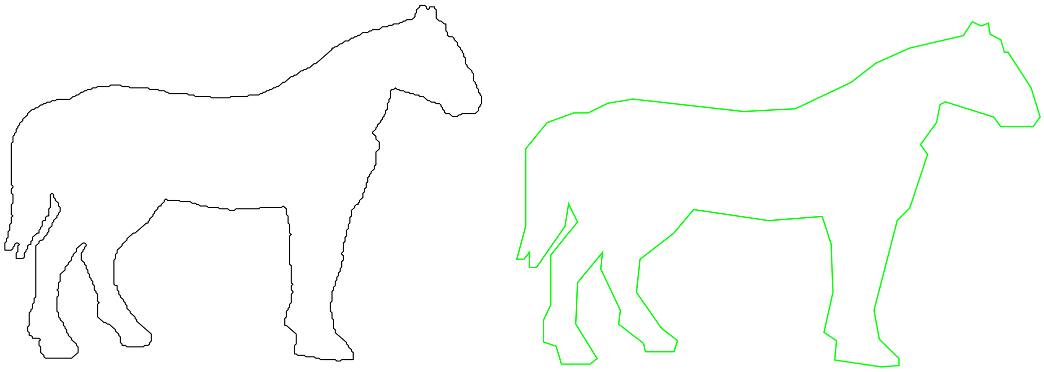
These systems apply different computer vision methods to recognize objects, areas of interest in digital images, determine the configuration of some object to automatize an industrial process, etc. We focus on different items of digital images and information sources depending on the task that the system is performing.

1.1 Polygonal approximations

One of the main characteristics used in Computer Vision is the shape of the objects. The contour of an object contains a great amount of information that we can use for a variety of purposes. However, the shape of an object also contains redundant data that can be removed without devaluing the original information. This can be done by reducing the size of the representation of the shape. This compression is achieved by approximating the shape of the contour using line segments. This reduced version of the contour is known as polygonal approximation in the literature.

Formally, we can define a polygonal approximation of a curve A as a subset of points M of the original set of points N of the curve. The size of the subset of points M is always lower (or equal) than the size of the original set of points N. Therefore, the polygonal approximation is the result of connecting the consecutive points of the subset M. In Fig. 1.1 a contour of a horse and an example of polygonal approximation are shown. Polygonal approximations are usually formed by line segments, however, other models are present in the literature [30, 42, 11, 36, 56].

As is said above, the main purpose of the polygonal approximation is to reduce the amount



(a) Contour of a horse (b) Polygonal approximation

Figure 1.1: This figure shows an original contour of a horse with 1383 points in (a) and a polygonal approximation of 75 line segments in (b).

of information needed to represent the shape of an object. For instance, in the literature we can find these methods for reducing the amount of information in several fields: medical image analysis [21], industrial robots garment manipulation [51], preprocessing of other computer vision methods [50, 62]. However, we can use polygonal approximations of digital planar curves for other purposes like object detection and shape recognition [29, 35, 52].

1.1.1 Assessment of the quality of polygonal approximations

In the literature there are a great amount of methods to obtain polygonal approximations. However, the evaluation of the quality of the approximations and the performance of the algorithms have demonstrated to be a complex task.

Most interest in assessing polygonal approximations focus on quantifying the physical distortion regarding the original planar curve. Different distortion measures can be found in the literature. A widely used distortion measure is the known as Integer Square Error (ISE). Let us suppose that a curve S has been approximated using a segment $\overline{s_i, s_j}$, then a distortion measure can be defined as

$$\Delta(i,j) = \sum_{k=i}^{j} d(s_k, \overline{s_i, s_j})^2 \tag{1.1}$$

where $d(s_k, \overline{s_i, s_j})$ is the orthogonal distance from the point s_k to the segment $\overline{s_i, s_j}$. Therefore, the term $\Delta(i,j)$ is the summation of the squared distance of the points between s_i and s_j and the segment $\overline{s_i, s_j}$. The summation of these distortions for all the line segments of the polygonal approximation give us the Integral Square Error, also known as the L_2-norm in the literature. Some authors use the maximum orthogonal distance from a point s_k to the segment that approximates it. This distortion measure is known as E_∞ or L_∞-norm in the literature.

These are the most common distortion measures used to solve the polygonal approximation problem, however some authors have used other distortion measures to solve the polygonal approximation problem. For instance, instead of using the squared orthogonal distance (L_2-norm), Pikaz and Dinstein [39] used the absolute orthogonal distance (L_1-norm or city block metric).

As explained above, one of the main uses of polygonal approximations is the data reduction on planar curve representation. For this reason, one of the first assessing measures defined was the compression ratio (CR), defined as

$$\mathrm{CR} = \frac{N}{M} \tag{1.2}$$

where N is the number of points of the original planar curve and M is the number of points of the polygonal approximation obtained.

Based on the Integral Square Error and the Compression Ratio, Sarkar [47] proposed to combine these two measures in a normalized Figure Of Merit (FOM) defined as

$$\mathrm{FOM} = \frac{\mathrm{CR}}{\mathrm{ISE}} \tag{1.3}$$

This measure of evaluation is biased towards polygonal approximation with lower distortion as was demonstrated by Rosin [43]. We can obtain polygonal approximations with a low value of distortion associated by obtaining solutions with high number M of segments. Moreover, this evaluation measure is not suitable for comparing polygonal approximation with a different number of segments.

Marji and Siy [24] proposed a modified version of the FOM defined as

$$F_x = \frac{\mathrm{ISE}}{\mathrm{CR}^x} \tag{1.4}$$

where x is used to reduce the imbalance between the numerator and the denominator. The common values used for x are 1, 2 and 3. Carmona-Poyato et al. [6] demonstrated that the best performance is obtained when $x = 2$ because numerator and denominator are of equal power.

Other modification of the FOM was defined by Nguyen and Debled-Rennesson [31]. The authors combines the ISE, the maximum deviation and the number of line segments of the polygonal approximation to assess the solution. This measure is defined as

$$\mathrm{MFOM}_3 = \mathrm{ISE} \cdot \mathrm{E}_\infty \cdot \mathrm{M}^3 \tag{1.5}$$

where M is the number of line segments of the solution.

In [19] the parametric FOM was proposed. This modified FOM is defined as follows

$$\mathrm{FOM}_a = \mathrm{ISE} \cdot \mathrm{M}^a \tag{1.6}$$

where the parameter a is computed for each contour. This is done by using the distortion-ration curve (RD) that depends on the values selected for the lower (M_1) and upper (M_2) bounds of the number of segments (M) of the solution.

Other modified version of the FOM was defined by Kolesnikov and Kauranne [20]. This measure is defined as

$$\text{pFOM}_p = \begin{cases} \text{ISE} \cdot \text{M}^a, a = 2D_2^{-1}, p = 2 \\ \text{E}_\infty \cdot \text{M}^a, a = 2D_\infty^{-1}, p = \infty \end{cases} \tag{1.7}$$

where M is the number of line segments of the polygonal approximation, and the parameter D_p is obtained from a log-linear model.

Rosin [43] showed that the measures that combine ISE and CR are not suitable to assess polygonal approximations with a different number of points. In [43], the author proposed a novel framework to assess the quality of polygonal approximations based on two measures: fidelity and efficiency. Fidelity is defined as

$$\text{Fidelity} = \frac{E_{\text{opt}}}{E_{\text{approx}}} \times 100 \tag{1.8}$$

where E_{approx} is the distortion of polygonal approximation which is been evaluated and E_{opt} is the distortion of the optimal polygonal approximation with the same number of points.

Efficiency is defined as

$$\text{Efficiency} = \frac{M_{\text{opt}}}{M_{\text{approx}}} \times 100 \tag{1.9}$$

where M_{approx} is the number of segments of the polygonal approximation and M_{approx} is the number of segments that an optimal polygonal approximation would require to obtain the same error.

These two measures are combined using a geometric mean known as Rosin's merit. This merit is defined as

$$\text{Merit} = \sqrt{\text{Fidelity} \times \text{Efficiency}} \tag{1.10}$$

The main advantage of using this merit function to assess polygonal approximations is, that can be used to compare approximations with different number of segments in a fair way. However, this framework to assess the quality presents some problems. Carmona-Poyato et al. [8] stated that this framework does not take into account if the number of segments of the solution to be evaluated is adequate or not. For instance, an optimal polygonal approximation with very few line segments (e.g. $M = 3$) will obtain better merit than a suboptimal solution with a reasonable number of line segments.

To deal with this problem a novel framework for evaluating the quality of polygonal approximations was proposed in [8]. This method obtains an optimal polygonal approximation that is used as a reference (PA_{ref}) by using a thresholding method. This optimal solution has a distortion ISE_{ref} associated and a number of points n_{ref}. Then, for each optimal polygonal approximation PA(i) with a distortion ISE(i) associated and a number of points n(i), a value $M(i)$ is defined as

$$M(i) = 100 \cdot (1 - \frac{|n(i) - n_{ref}|}{n_b}) \cdot e^{-\frac{|\text{ISE}_{ref} - \text{ISE}(i)|}{min(\text{ISE}_{ref}, \text{ISE}(i))}} \tag{1.11}$$

where n_b is the number of breakpoints of the original planar curve. The authors obtain the efficiency curve that represents the values of $M(i)$ versus the number of points $n(i)$. In a similar way the fidelity curve that represents the values $M(i)$ and the distortion ISE(i) is obtained. Using these two curves we can estimate a value of efficiency and fidelity using the number of points and the distortion (ISE) of the polygonal approximation to be evaluated.

Most of the measures use a combination of the distortion (ISE) and the compression ration (CR). However, some other distortion measures are used to assess a polygonal approximation. For example, Parvez and Mahmoud [33] defined a measure which combines the length of the polygonal approximation and the original contour. The measure is defined as

$$\text{LR} = \frac{L_d}{L} \tag{1.12}$$

where L_d is length of the solution and L is the length of the original contour. Lowe [23] proposed to use the maximum deviation (E_∞) of the solution and the length of the original shape to determine the significance . This measure is defines as follows

$$\frac{L}{E_\infty} \tag{1.13}$$

Sato [48] defined a distortion measure based on the ration between the length of the solution and the original length. This measure was defined as

$$\varepsilon = \frac{L - L_d}{L} \tag{1.14}$$

Other authors [55] have proposed to use the deviation of the normalized area as

$$\frac{|A - A_p|}{L} \tag{1.15}$$

where A is the area of the original contour, A_p is the area of the polygonal approximation and L is the length of the contour.

1.1.2 State of the art

The polygonal approximation problem is usually defined in two separated ways as was stated in [18]. Depending on the function to minimize the polygonal approximation problem can be defined as:

- min-#: In this problem the number of line segments M that forms a polygonal approximation are minimized. The distortion error should not excess a threshold ε defined by the user. The optimal solution to this problem should also have the minimum distortion associated among all the solutions with the same number of line segments.

- min-ε: This optimization problem minimizes the distortion associated to a polygonal approximation with a number of line segments M fixed by the user.

Several method have been proposed in the literature to solve both the min-ε and the min-# problems. Depending on the task we are carrying out a different distortion measure may be used: L_∞-norm is used to assure that the maximum deviation does not exceed a threshold determined by the user; L_2-norm is used to obtain a polygonal approximation whose distortion is lower than a threshold.

We can separate the different methods to obtain polygonal approximations depending on the optimality of the solution found. Thus, algorithms can be classified into optimal and suboptimal methods.

Optimal methods

As is explained above, the polygonal approximation problem is a discrete optimization problem. Thus, the optimal methods to solve, both the min-ε and the min-# problems, are based on different optimization frameworks.

We can consider the first optimal algorithm the method introduced by Papakonstantinou [32]. This method based on Dynamic Programming, computes the minimum number of segments needed to approximate the points from P_i to P_j, taking into account that the maximum deviation is lower than the error threshold supplied by the user.

One of the first optimal algorithm described in the literature was proposed by Dunham [13]. This approach uses the L_∞-norm as the distortion criterion for solving the min-# problem. The method described is based on the recursive computation of the minimum number of segments needed to approximate the original open curve C from the initial point P_0 to a point P_u. This computation is done using the Dynamic Programming framework.

Another optimal alternative for solving the min-# problem was proposed by Melkman and O'Rourke [28]. The authors sort out the problem using the L_∞-norm by constructing a graph where the nodes are the vertices of the curve C. Nodes for points P_i and P_j are connected by an arc if the maximum error for the approximated segment $\overline{P_iP_j}$ is lower than the defined error threshold.

An interesting approach was proposed by Sato [48]. The author obtains the optimal polygonal approximation with a number of segments M which minimizes the differences of the arc length of the original curve and the approximation. That is, the optimal solution is the polygonal approximation with M segments and the maximum arc length.

Perez and Vidal [37] proposed to solve the min-ε problem using the L_2-norm distortion measure. This method based on Dynamic Programming, computes all the possibilities to approximate point P_i using M segments. This process is repeated until the last point is reached. A method to compute the distortion values in constant time $O(1)$ is also given. The definition of the algorithm appears in Algorithm 1.

Data: C (Digital planar curve), N (Number of points of the curve), M (Number of segments of the solution)
Result: The optimal polygonal approximation
var g // *used to memorize the minimum global error to reach any point of the contour using any number of segments*;
var Points // *Points of the digital planar curve*;
var Father // *Array that contains the ending point of the previous segment*;
$g[1,0] \leftarrow 0$;
for $n \leftarrow 2$ **to** N **do**
    $g[n,0] \leftarrow$ maxValue;
end
for $m \leftarrow 1$ **to** N **do**
    for $n \leftarrow 2$ **to** M **do**
        // Search the minimum error to reach point n with m segments;
        $g[n,m] \leftarrow \min_{i \in [m,n-1]} g[i,m-1] + \text{error}(i,n)$ // Memorize i_{min};
        Father$[n,m] \leftarrow i_{min}$;
    end
end
TotalError $\leftarrow g[n,m]$;

Algorithm 1: Algorithm to solve optimally the min-ε problem proposed by Perez and Vidal [37].

Another alternative based on Dynamic Programming was proposed by Tseng et al. [54]. This author used three different error measures to optimally solve the min-# problem. The problem is sorted out using the L_∞-norm, L_2-norm and also the differences of the perimeter between the original curve and the polygonal approximation.

A different approach based on graph search was introduced by Salotti [44]. The novel method uses the $A^\star$ algorithm to search for the optimum solution of the min-ε problem using the L_2-norm error criterion. The nodes of the graph represent the ending points of the segments and their rank (the number of segments needed to reach that point). An edge between two nodes defines a line segment, where the cost, the error value of the segment, is associated to that edge. The function $f(x)$ associated to a node is formed by the function $g(x)$ plus the heuristic function $h(x)$. Function $g(x)$ represents the cost (distortion) required to reach point of the node x from the initial point. The heuristic function $h(x)$ represents a lower bound of the remaining distortion to reach the ending point of the curve. The defined heuristic function is computed using the linear regressions of the points between the point of node x and the ending point of the curve.

Salotti [45] also solved the min-# optimally using the L_2-norm error criterion. This approach also uses the $A^\star$ algorithm to obtain the optimal solution. To compute the $g(x)$ function a creation cost is added to sort the solutions according to two criteria: the number of segments and the distortion error associated. The heuristic function $h(x)$ represents a lower bound of the remaining number of segments to reach the ending point of the curve from the point of node x. To compute this lower bound a procedure that uses the linear regression is defined.

Most of the methods proposed to optimally solve both the min-ε and the min-# problem, are defined for open curves. To optimally solves these problems on closed curves have been proposed to duplicate the initial point of the curve as also the ending one. Then, the algorithm must try all points of the curve as the fixed point of the final solution [37]. This solution can be applied on small contours, due to the huge computational burden needed.

To overcome this problem, some heuristic methods have been proposed. Sato [48] proposed to use the farthest point to the centroid of the curve as one of the distinctive points. Horng and Li [15] proposed a method to determine a good candidate for using as the fixed point of the final solution. The method first select a random fixed point for the final solution and executes the Dynamic Programming algorithm. Taking into account the vertices obtained after run the Dynamic Programming process, the method selects a new initial point for the second iteration. This new initial point is the farthest vertex from the initial point of the first iteration and is separated from its nearest vertex by more than a given threshold. The authors suggest a value for the threshold equal to $\frac{0.5N}{M}$, where N is the length of the shape and M is the number of points of the polygonal approximation. This two iterations algorithm is heuristic and therefore may lead to suboptimal solutions.

Suboptimal methods

The number of methods to solve the polygonal approximation problem is enormous. To give an overview of the methods used in the literature an small classification is proposed. Among the huge number of algorithms proposed, we can separate them into two categories: methods based on metaheuristics and heuristics.

Metaheuristics have been widely used to successfully solve the polygonal approximation problem. For instance, the Ant Colony Optimization (ACO) framework [65] has been used to solve the min-# problem using the L_2-norm. This method represents the problem using a graph where nodes are the points of the curve and the edges between two nodes represent the approximation error associated to the segment. The ants are initially placed in some nodes and move from node to node until a complete tour is completed, that is, the initial node is reached again. The pheromone intensity rule used in this approach is computed used the quality of the solution obtained in each iteration. The polygonal approximation returned is the best solution found at a specified iteration.

Another metaheuristic approach based on the Artificial Bee Colony (ABC) was proposed by Huang [16]. This nature inspired method solves the min-ε problem using the L_2-norm. The process starts generating a set of solutions. A fitness function is compute for each solution. A serch process is then executed, where different stages (Employed Bee, Onlooker Bee and Scout Bee Stage) try to optimize the pool of solutions generated. If a solution is improved, that is, the fitness function is better than the previous solution; is stored in the pool. Otherwise, the old solution is kept and the trial is incremented. If the trial of any solution is higher than a limit, then this solution is discarded. This search process is repeated a number of times and then the best stored solution is returned.

Another optimization framework used to solve the polygonal approximation problem is the Particle Swarm Optimization approach. Yin [66] was the first author to solve the min-# problem by using this framework of optimization. The representation of each particle (possible solution) is done by a binary vector, where each item is one or zero whether this vertice is selected for the solution. A fitness function is used to assess the quality of the solution in the optimization process. This process determines the best solution for each particle and for the whole swarm at each iteration. The process is repeated a number of iterations and then the best solution is returned. To increment the diversity of the population of the particles, Wang et al. [57] incorporate a mutation operator. Wang et al. [61] solves the min-ε and min-# problem using the PSO optimization framework building a probabilistic model of the distribution of the good regions in the search space (Estimation of Distribution Algorithms). Wang et al. [60] proposed a method to solve the min-# problem. This method uses the Integer Particle Swarm Optimization framework, where the particles are represented using integer coding scheme.

Genetic Algorithms have also been used to solve the polygonal approximation problem. A first solution to the min-ε problem was proposed by Yin [64]. The individuals (solutions) are represented by binary strings, where a bit is equal to one if this vertice is select for the final solution, or zero otherwise. A fitness function represent how well a solution approximates the original planar curve. In each iteration this population is evolved using the defined mutation operator and using a process of elitism. We can find other approaches [17, 67, 46, 22, 14, 59] that employ the genetic algorithm framework using different distortion measures and types of polygonal approximation problems.

Other metaheuristic approaches have been used to solve the polygonal approximation problem. For instance, Zhang and Guo [67] proposed a tabu search approach to solve the min-# using the L_2-norm as the distortion measure.

The main drawback of the metaheuristic approaches is the computational cost. Usually, heuristics approaches obtain worse results than the metaheuristic approaches, but the computation burden is lower. The great number of heuristic methods described in the literature can be classified into three different subcategories: sequential scan approach, split approach, merge approach.

The sequential scan approach is a fast type of polygonal approximation framework, where the algorithm performs an operation on each point of the curve. Using the result of this operation the method is able to create a polygonal approximation. This type of algorithms are very fast because the complexity order is $O(N)$. Some examples of this kind of algorithms could be found in the literature [49, 53, 41, 24]

Some methods start using a rough approximation as the initial solution for the polygonal approximation of the curve C. The methods keep adding points to this initial approximation until some criteria is satisfied, for instance, the distortion error is lower than a threshold. This heuristic is known in the literature as the split approach and some examples could be found in [40, 12, 23].

On the other hand, several methods use an opposite approach, that is, the initial solution is the whole curve C. These algorithms merges line segments deleting the intermediary points of these segments. This procedure is repeated until the condition is violated, for instance, the distortion error associated is higher than a threshold. This merge approach is widely used in the literature [38, 27, 25, 26, 7].

These heuristic approaches are usually put together into the split-and-merge approach. This type of algorithms start with an initial polygonal approximation. Then, the two stages split-and-merge process is iteratively executed. First the segment with the maximum distortion error associated to the point P_k is split into two segments by adding this point P_k to the solution. Then, the merge procedure searches for the point P_i with the minimum distortion to the segment connecting its two adjacent points $\overline{P_{i-1}, P_{i+1}}$, and removes P_i from the solution. This process is graphically shown in Fig. 1.2. This procedure repeats until a condition is satisfied. This two stages procedure is very popular and several examples can be found in the literature [34, 5, 63, 58].

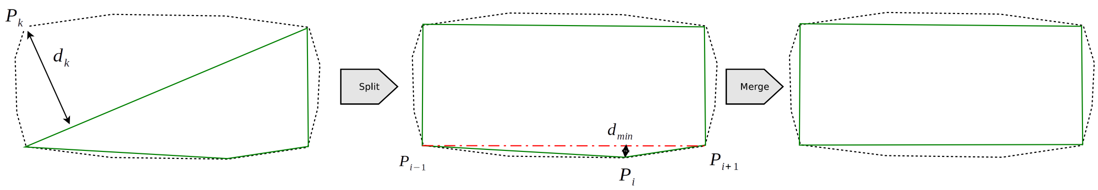


Figure 1.2: Split and merge heuristics are usually used together to solve the polygonal approix-mation problem.

Part II

Contributions

Chapter 2

First contribution: The computation of polygonal approximations for 2D contours based on a concavity tree

J. Vis. Commun. Image R. 25 (2014) 1905–1917

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

The computation of polygonal approximations for 2D contours based on a concavity tree ☆

E.J. Aguilera-Aguilera, A. Carmona-Poyato *, F.J. Madrid-Cuevas, R. Medina-Carnicer

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

ARTICLE INFO

Article history:
Received 3 February 2014
Accepted 18 September 2014

Keywords:
Digital planar curves
Polygonal approximation
Dominant points
Concavity tree
Breakpoints
Merge methods
Split methods
Convex hull

ABSTRACT

In this work, a new proposal to improve some methods based on the merge approach to obtain polygonal approximations in 2D contours is presented. These methods use a set of candidate dominant points (*CDPs*) to obtain a polygonal approximation. Then, redundant candidate dominant points of the set of *CDPs* are deleted, and the remaining dominant points will be the polygonal approximation of the original contour. The main drawback of most of these methods is that they use all breakpoints as *CDPs* and most of these breakpoints depict only the noise of the original contour.

Our proposal, based on a concavity tree, obtains a more reduced and significant set of *CDPs*. When this proposal is used by some methods based on the merge approach (the Masood methods and the Carmona method), their computation times are reduced. The experimental results show that the new proposal is efficient and improves the tested methods.

1. Introduction

Polygonal approximation can be categorised into two classes of subproblems [15]:

(1) *Min* $-\#$ problem: given an *N*-vertex polygonal curve *P*, approximate this curve by using another polygonal curve *Q* with a minimum number of segments *M*, considering that the approximation error *E* does not exceed a given maximum tolerance E_0.
(2) *Min* $-\epsilon_0$ problem: given an *N*-vertex polygonal curve *P*, approximate this curve by using another polygonal curve *Q* with a given number of line segments *M*, so that the approximation error *E* is minimised.

The proposed methods to solve these problems can be classified as heuristic, metaheuristic, and optimal algorithms.

- Heuristic algorithms produce non-optimal polygonal approximations with a low computational cost. Heuristic algorithms are based on a greedy approach and can be classified as follows:
  - Sequential scan approach [36,29].
  - Split approach [28,9,23].
  - Merge approach [26,17,20–22,7].
  - Split and merge approach [24,18,42,12].
- Metaheuristic approach with different stochastic optimisation methods:
  - Genetic algorithms [13,39].
  - Colony optimisation [45].
  - Particle swarm optimisation [38,40].
  - Tabu Search [44,45].
- Optimal algorithms produce optimal polygonal approximations but a high computational cost is required. Usually optimal algorithms are relied in Dynamic Programming [2,10,25,27,15] or an A^*-search approach [32,33].

According to Attneave [1], most of the information in a contour is located at points of high curvature. For this reason, these points are used to obtain polygonal approximations. These points are known as dominant points and are an important target in many machine vision applications [41]. The dominant points are used in image matching, shape description and pattern recognition because they provide significant data reduction while preserving crucial information about the object [14]. The dominant points are used in computer vision, for example, in an aerial image, and dominant points are used to recognise man-made objects. In a time sequence, dominant points can be used to compute the displacement between each pair of consecutive images, etc. [14].

☆ This work has been developed with the support of the Research Projects called TIN2012-32952 and BROCA both financed by Science and Technology Ministry of Spain and FEDER. We thank the reviewers for their valuable contributions to improve this work.

* Corresponding author. Fax: +34 957218630.
E-mail address: ma1capoa@uco.es (A. Carmona-Poyato).

http://dx.doi.org/10.1016/j.jvcir.2014.09.012
1047-3203/

A candidate dominant point (*CDP*) is defined as a high-curvature point that belongs to the set of candidates to be used to obtain a polygonal approximation. This candidate can be selected or removed according to the method used for obtaining the polygonal approximation. The breakpoints of a contour are points where the contour turns. To be more exact, a point is a breakpoint if its Freeman chain code is not equal to the chain code of its previous point. Because the dominant points are points of high curvature, the dominant points are all breakpoints. Therefore, dominant points are breakpoints that provide significant and important information about the contour, and they are connected to obtain a polygonal approximation of the original contour. For this reason, all the breakpoints of a contour are used as *CDPs* to obtain its polygonal approximation in most merge methods [26,20–22,7]. However, the number of breakpoints is very high in any contour, and most of them are redundant candidate dominant points for two reasons:

- They may depict the noise obtained by the digitization of the original contour.
- They may belong to a digital straight segment.

For these reasons most of the breakpoints never belong to the polygonal approximation.

These redundant candidate dominant points are then deleted, and only the most significant and important candidates are selected to obtain the polygonal approximation. Furthermore, the computation time of these methods can be reduced if the set of candidate dominant points is reduced by deleting the redundant candidate dominant points.

In this paper, a new proposal, based on a concavity tree, to reduce and improve the initial set of candidate dominant points is presented. An extended concavity tree (different from the original concavity tree proposed by Sklansky [35]) is obtained. From this extended concavity tree, convex hulls, corresponding to different parts of the contour with different levels of detail, are obtained. These convex hulls may represent concave and convex parts of the contour. The endpoints corresponding to concave and convex parts are significant and important dominant points for the following reasons:

- They are breakpoints.
- They can be used to separate the concave and convex parts of the contour.
- They consider the different levels of detail of the different parts of the contour. In this way, fewer dominant points are added where the contour is straight or relatively straight and more dominant points are added where the contour is curved.

For these reasons, these endpoints can be used as *CDPs*.

The new proposal is explained in detail in Section 4.

Section 2 describes the calculation of the original concavity tree and the previous methods related to the present proposal. Section 3 presents some basic measurements to evaluate the quality of a polygonal approximation. Section 4 describes how to obtain the extended concavity tree for the present proposal, and Section 5 discusses the performance of the proposal. Finally, Section 6 shows the main conclusions.

2. Related work

A concavity tree is a tree whose different levels represent the convex and concave parts of a contour. These convex and concave parts are obtained using the convex hull. In a 2D contour, the convex hull is the smallest convex polygon that contains all points of the contour.

The concave parts (concavities) are formed by points of the contour that do not match the vertices of the convex hull and do not belong to the segment lines that join two consecutive points of the convex hull.

The convex parts are formed by points of the contour that match the vertices of the convex hull or belong to the segment lines that join two consecutive points of the convex hull.

When all points of the original contour belong to the segment lines that join two consecutive points of the convex hull, there are no concave parts, and the convex hull represents the original contour faithfully. In this case, the tree contains only the root. In the other case, there are concave parts, and these concave parts can be obtained from the points of the contour that do not belong to the segment lines that join two consecutive points of the convex hull. These concave parts will be the sons of the root.

For each concave part there are two vertices, the previous and next vertices belonging to the convex hull. The segment line that joins these vertices is called the cover. If we consider the points of a concave part and the ends of its cover as a subcontour corresponding to this concave part, a new convex hull of this subcontour can be obtained. Thus, the subcontour can be treated as the original contour obtaining new concave parts and their corresponding nodes are added to the concavity tree, etc. Finally, the concavity tree represents the original contour faithfully. (See Fig. 1.)

To test if a contour or subcontour has concave parts, the sum of the squared orthogonal distances between the contour or subcontour points and their corresponding segment line that joins their previous and next points of the convex hull, is used. If this sum is not 0, the contour or subcontour has concave parts. For this reason, the measurement *ISE* is used. This measurement is explained in Section 3.

Fig. 1 shows a synthetic contour (a), its concavities (b, c, d, e, f, g) and its concavity tree depicting the hierarchies between its concavities (h). In this case, on the first level, the whole contour is represented (a). The boundary has four concavities represented as subcontours in the level two of the tree. On the second level, the first subcontour (b) and the last subcontour (e) are convex, and they have no concavities, the second subcontour (c) and the third subcontour (d) are not convex and have one triangular concavity (third level, (f) and (g)). In (h), the hierarchies between the concavities are shown. The nodes are expanded until no new concavity is obtained.

The concavity tree was first introduced in [35], then the concavity has been proposed to represent 2D shapes [3–5,43,11,31,16]. In these studies, the concavity tree has been used as a representation of the original contour or to obtain minimal length polygons.

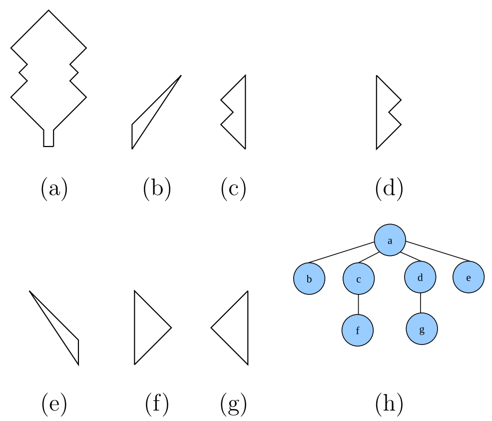


Fig. 1. Example of concavity tree of a synthetic contour. Original contour (a). Subcontours corresponding to the concavities of first level (b), (c), (d) and (e). Subcontours corresponding to the concavities of second level (f) and (g). Concavity tree of the contour with the hierarchies between concavities (h).

In this paper, we use a extended concavity tree to obtain a significant and reduced set of *CDPs*. This set can be used by merge methods to obtain polygonal approximations.

The proposal presented will be used with the Masood [20–22] and Carmona [7] methods to improve them.

Masood [20,21] proposed a method based on an iterative algorithm that deletes redundant dominant points with the smallest error value. Breakpoints are taken as the initial set of *CDPs*. To calculate the error associated with each dominant point DP_j, two neighbouring dominant points, DP_{j-1} and DP_{j+1}, are joined with a straight line. The maximum perpendicular (squared) distance of all boundary points between DP_{j-1} and DP_{j+1} from the straight line is called the associated error value (*AEV*) of dominant point DP_j. In each iteration, only the redundant dominant point with the smallest error value (AEV_{min}) is eliminated. A stop condition is used to end the iterative elimination. Redundant dominant points are deleted until the error due to the deletion is greater than a fixed error, ϵ.

Masood [22] proposed an optimised and related method. This method is based on the suppression of redundant dominant points from an initial set of candidate dominant points (breakpoints) with the smallest error value (AEV_{min}) and then, by using an optimisation and adjust procedure, the method reduces the integral square error (*ISE*) when a redundant dominant point is deleted. The integral square error (*ISE*) can be reduced if the neighbouring points share the error properly when a redundant dominant point, DP_j, is eliminated. For this purpose, some *DP*s may adjust their position. When a *DP* is moved, that *DP* may de-optimise the position of the two neighbouring *DPs*. Therefore, an optimisation algorithm must be iterated until all of the *DPs* are optimised. Masood therefore recommended updating the associated error value (*AEV*) of neighbouring *DPs* (in sequence on both sides) until the new *AEV* of any *DP* is equal to its previous *AEV*. Masood compared the polygonal approximations obtained using his second method so that the number of *DPs* was equal to the *DPs* of the compared algorithm. The results were all optimal. The complexity of this method is $O(MN^2)$, where M is the number of points of the polygonal approximation and N is the number of points of the original contour.

The Masood methods [20–22] delete one redundant dominant point in each iteration. For this reason, a polygonal approximation with any number of points could be obtained.

Carmona et al. [7] proposed a method to obtain polygonal approximations based on the suppression of dominant points from an initial set of candidate dominant points (breakpoints). The method of Carmona removes redundant dominant points until a required level of approximation is obtained. The method is iterative. In each iteration, several quasi-collinear and redundant dominant points are suppressed, and a new polygonal approximation is obtained. In this case, polygonal approximations with any number of points cannot be obtained.

3. Measurements to evaluate polygonal approximations

In this section, some basic measurements used in this work to evaluate the quality of polygonal approximations are described.

– Compression Ratio (*CR*).

$$CR = \frac{N}{M} \tag{1}$$

where N is the number of points in the contour, and M is the number of points of the polygonal approximation.

– Sum of square error (*ISE*). Given a segment with end points (P_i, P_j) the *ISE* for this segment is

$$ISE = \sum_{k=i+1}^{j-1} d_k^2 \tag{2}$$

where d_k is the orthogonal distance from point P_k to the straight line defined by points P_i, P_j. Fig. 2 shows a line segment (i, j) of the polygonal approximation corresponding to the points $i, 1, 2, 3, j$ of the original contour. In this case, $ISE = d_1^2 + d_2^2 + d_3^3$.

– Sarkar [34] combined these two measures as a ratio, producing a normalised figure of merit (*FOM*) defined as

$$FOM = \frac{CR}{ISE} \tag{3}$$

– Maximum error (E_∞) is the maximum value of d_k.

Rosin [30] shows that *FOM* is biased towards approximations with lower *ISE* because its two terms are unbalanced. He proposed a new measurement (merit) relying on two components: fidelity and efficiency. Fidelity measures how well the polygon obtained by the algorithm to be tested fits the curve relative to the optimal polygon in terms of the approximation error. Efficiency measures how compact the polygon obtained by the algorithm to be tested is, relative to the optimal polygon which incurs the same error. Depending on the shape of the curve, the two measures may vary considerably. Rosin used a combined measure (geometric mean of fidelity and efficiency).

$$Merit = \sqrt{Fidelity \times Efficiency} \tag{4}$$

Marji and Siy [19] used a modified version of *FOM* (in this case, he used the inverse of *FOM*). The new measure is defined as

$$F_x = \frac{ISE}{CR^x} \tag{5}$$

where x is used to control the contribution of the denominator to the overall result to reduce the imbalance between the two terms. They used $x = 1, 2$ and 3.

4. The present proposal

The objectives of the present proposal can be summarised as follows:

- Reduce the initial set of candidate dominant points.
- This initial set must consider the different levels of detail of the different parts of the original contour. In those parts where a high level of detail is needed, the number of candidate dominant points will be higher than in those parts where a small level of detail is needed. For example, in Fig. 3, a small level of detail is required between the points 1 and 14 because this part of the contour is more or less straight, and fewer dominant points are needed. However, a high level of detail is required between points 15 and 29 because this part of the contour contains more details, and a greater number of dominant points is needed.
- All relevant dominant points must belong to the initial set of *CDPs*. They are usually the points of transition between parts of different levels of detail. For example, in Fig. 3, some of the transition points are points 15, 29, 47 and 59 in the first level; points 22, 37, 42 and 53 in the second level; and points 16, 19, 25, 27, 49 and 51 in the third level.

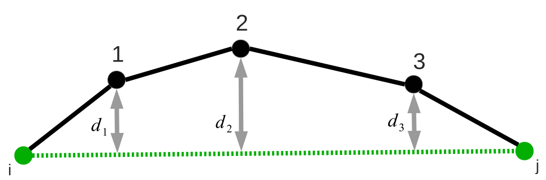


Fig. 2. Example of calculation of *ISE* for a line segment of the polygonal approximation.

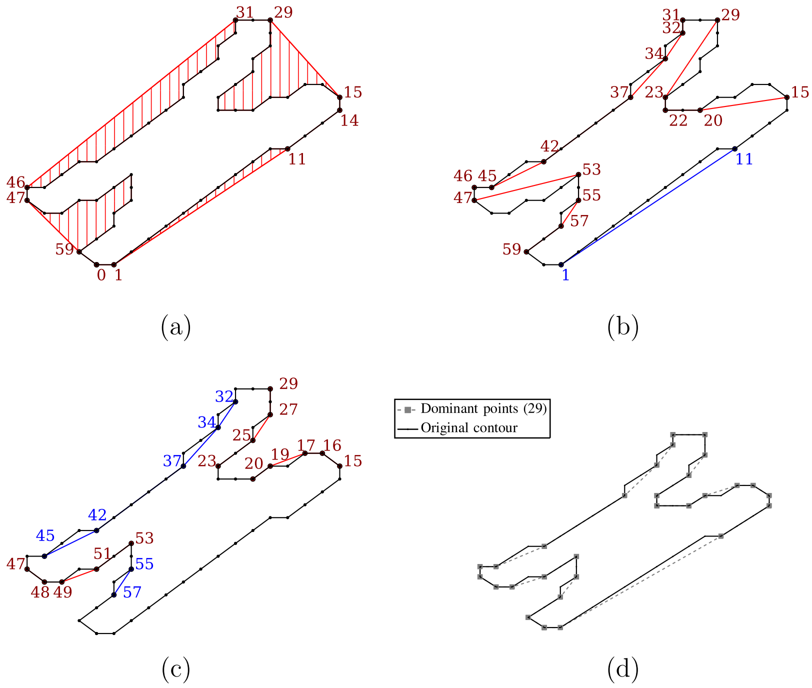


Fig. 3. Example of the proposed method for the chromosome contour. (a) First level polygonal approximation. (b) Second level polygonal approximation. (c) Third level polygonal approximation. (d) Set of final candidate dominant points.

To improve the quality of the initial set of *CDPs* and reduce the size of the set, a procedure based on the concavity tree is used. The nodes of the original concavity tree contain all the concavities along the contour and subcontours (Fig. 1). However, our proposal differs as follows:

- Each node (except the root) contains two consecutive vertices of the convex hull corresponding to the original contour or a subcontour. Thus, a node may represent the cover of a concave part (concavity) corresponding to the original contour or a subcontour.
- Only the nodes corresponding to the most relevant concavities are expanded. In the original concavity tree, all the concavities are expanded. For this reason, the extended concavity tree has fewer levels than the original concavity tree.

For this purpose, a split method is proposed. Fig. 3 (chromosome contour) is used as an example. In this Figure, to explain the method, the initial point is labelled with 0, and the contour will travel using the counterclockwise direction. The new points added in each level are labelled. The method is described in the next steps.

(1) The convex hull of the original contour is obtained. Fig. 3(a) shows the vertices of its convex hull. The convex hull consists of vertices 0, 1, 11, 14, 15, 29, 31, 46, 47 and 59. These points are added to the set of *CDPs*. All the segment lines that join two consecutive vertices are depicted in the nodes of second level of the extended concavity tree (Fig. 4). The first level depicts the original contour. Thus, the extended concavity tree will always have at least two levels.

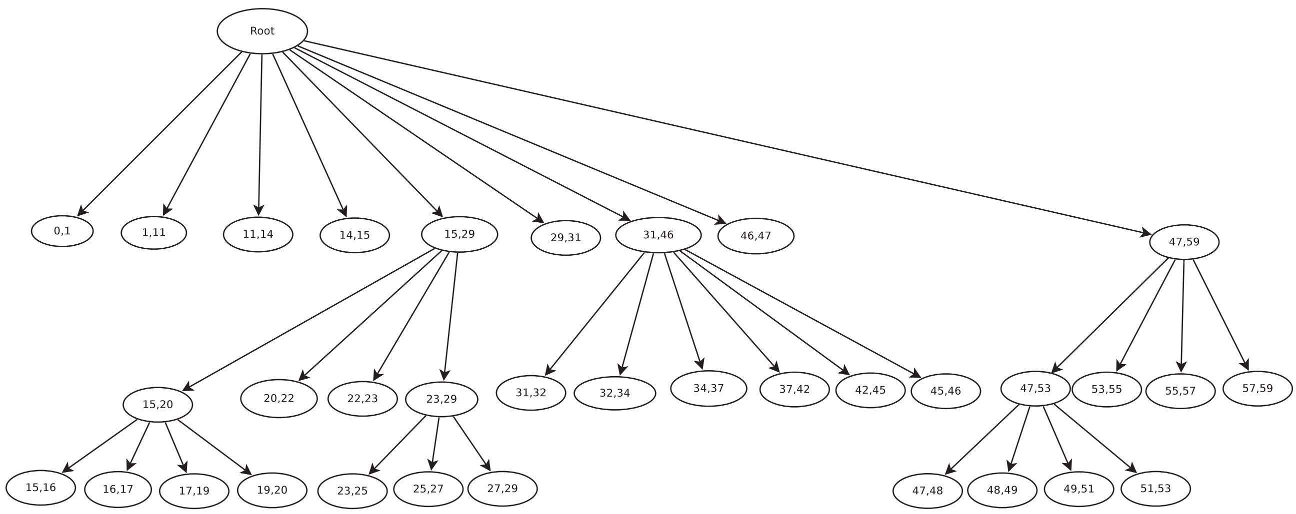


Fig. 4. Reduced concavity tree for chromosome contour.

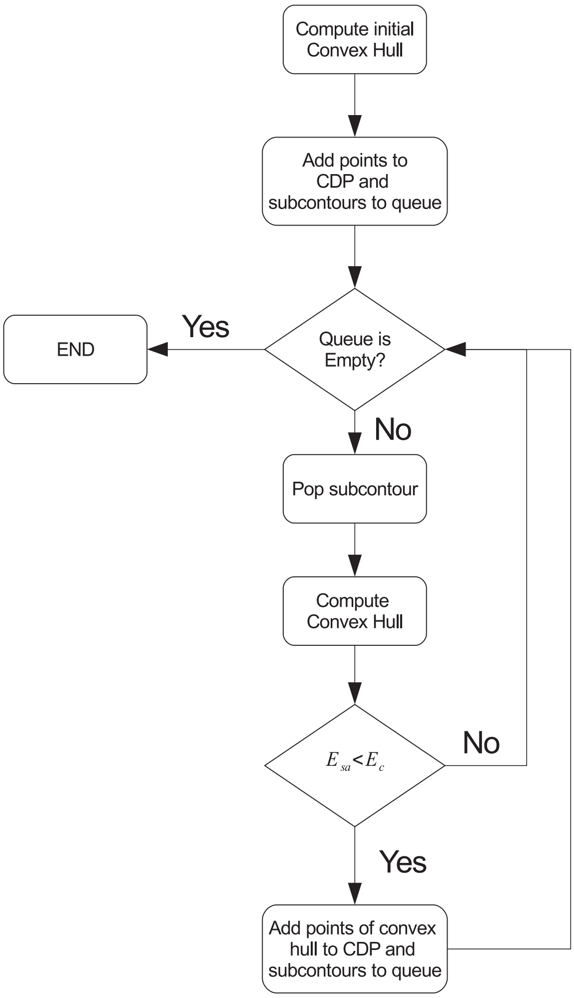


Fig. 5. Flowchart of the method to obtain the extended concavity tree and *CDP*.

(2) If the overall points of the original contour are contained in the segment lines that join the vertices of the convex hull, then there are no concave parts, and this convex hull represents the original contour faithfully. In the other case, there are concave parts, and each segment line depicted in the sons of the root is the cover of its concave part (if exists).

(3) If there are concave parts:

(a) The concave parts can be obtained from the points of the contour that do not belong to the segment lines that join two consecutive points of the convex hull. Fig. 3(a) shows four concave parts and its covers. In this case, the points between 1 and 11 are the first part, the points between 15 and 29 are the second part, the points between 31 and 46 are the third part, and the points between 47 and 59 are the last part. The ends of each cover and its corresponding concave part is considered a subcontour of the original contour. In this case, four subcontours, corresponding to shady zones in Fig. 3(a), are obtained.

(b) The *ISE*/*CR* value is calculated in each cover. This value is obtained considering that each cover is the polygonal approximation of its associated concave part. For example, cover (15, 29) is considered the polygonal approximation of its concave part. For this reason, the value of *ISE* is the corresponding value between the points 15 and 29, and $CR = 15/2$ because 15 is the number of points of concave part and 2 is the number of points of its polygonal approximation (its cover). This value is called the cover error (E_c), and it is a local error. In this case, *ISE* is divided by *CR* to weight the error with the number of points of the approximated segment. Thus, the values of E_c corresponding to the covers $(1,11), (15,29), (31,46), (47,59)$ are obtained.

(c) The convex hulls of the four subcontours are obtained. Considering that each subcontour is approximated by the segment lines that join the vertices of its corresponding convex hull, the values of *ISE*/*CR* are calculated for the four subcontours. For example, the subcontour corresponding to the points between 15 and 29 (15 points) is approximated by a convex hull with 5 vertices (15, 20, 22, 23 and 29), so the value of *ISE* is the *ISE* corresponding to the approximation using the convex hull and $CR = 15/5$. This value is called the subcontour approximation error (E_{sa}). Similarly to the calculation of E_c, *ISE* is divided by *CR* to weight the error with the number of points of the polygonal approximation.

(d) Now we have two values for each concave part:
- Cover error (E_c) considers the error obtained if the concave part is approximated by its cover, for example, the first concave part is approximated by the cover (15, 29).
- Subcontour approximation error (E_{sa}) considers the error obtained if the concave part is approximated using its convex hull, for example, the convex hull of the subcontour corresponding to the cover (15, 29) and its concave part.

If the E_{sa} value is less than the E_c value in a concave part, this part is approximated by the segment lines that join the vertices of its corresponding convex hull, and their vertices are added to the set of *CDPs*. Thus, the node corresponding to the associated cover is expanded, and nodes corresponding to consecutive vertices of the convex hull of the subcontour are added (nodes of the third level in Fig. 4). Otherwise, this concave part will be

Table 1
Obtained results for all the used contours. *N* is the number of the points of the original contour. N_{BP} is the number of initial candidate dominant points for the related methods. N_{CDP} is the number of initial candidate dominant points for the proposed method. N_{nodes} is the number of nodes of the extended concavity tree. N_{levels} is the number of levels of the extended concavity tree.

Contour	N	N_{BP}	N_{CDP}	N_{nodes}	N_{levels}
tin-openers	580	242	89	108	4
pliers	2040	1020	187	216	5
plane1	1015	477	125	145	5
plane2	787	397	121	149	4
plane3	1073	535	129	152	5
plane4	1126	461	160	192	4
plane5	1098	579	158	188	4
plane6	921	483	128	148	4
plane7	897	460	114	128	4
rabbit	745	334	147	175	5
chromosome	60	36	29	35	3
spoon	1370	1070	76	91	4
screwdriver	1677	1410	76	90	4
dinosaur1	795	307	121	144	4
dinosaur2	625	263	111	139	5
dinosaur3	889	379	156	190	5
dinosaur4	779	298	143	174	5
leaf	120	56	37	51	3
infinity	45	30	18	24	4
hand	1041	370	168	219	5
hammer	1583	1017	66	74	3
semicircles	102	52	29	36	3
turtle	553	273	109	132	5

approximated using its cover, and no nodes will be added to the extended concavity tree. In the example, only three concave parts (corresponding to covers (15,29), (31,46) and (47,59)) are approximated using their convex hulls (red lines in Fig. 3(b)). The first concave part (corresponding to cover (1,11)) is approximated using its cover (blue line in Fig. 3(b)).

(e) This procedure is applied to the concave parts approximated by the convex hull in a similar way, until no E_{sa} value is better than its corresponding E_c. Thus, the set of *CDPs* is obtained. Fig. 3(c) shows the next level of polygonal approximation and the new dominant points added. In this case, the method has been applied to the three concave parts approximated by the convex hull and obtained in the previous level. For example, the first subcontour approximated by the convex hull in the previous level (corresponding to cover (15,29)) is split into two subcontours (subcontour between points 15 and 20 and subcontour between points 23 and 29), and their corresponding concave parts are approximated using their convex hulls. Thus, five new dominant points (16, 17, 19, 25 and 27) are added to the *CDPs* ((see Fig. 3(c)). In a similar way, the points 48, 49 and 51 are added at this level.

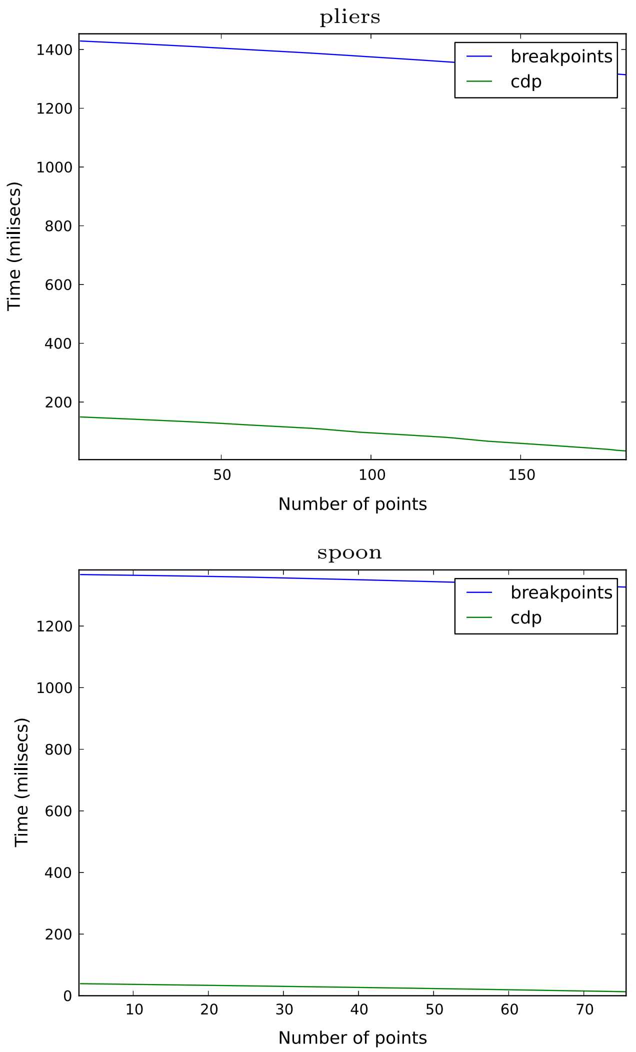


Fig. 6. Times obtained by the original Masood method (*breakpoints*) and the modified Masood method with our proposal (*CDP*), using pliers and spoon contours.

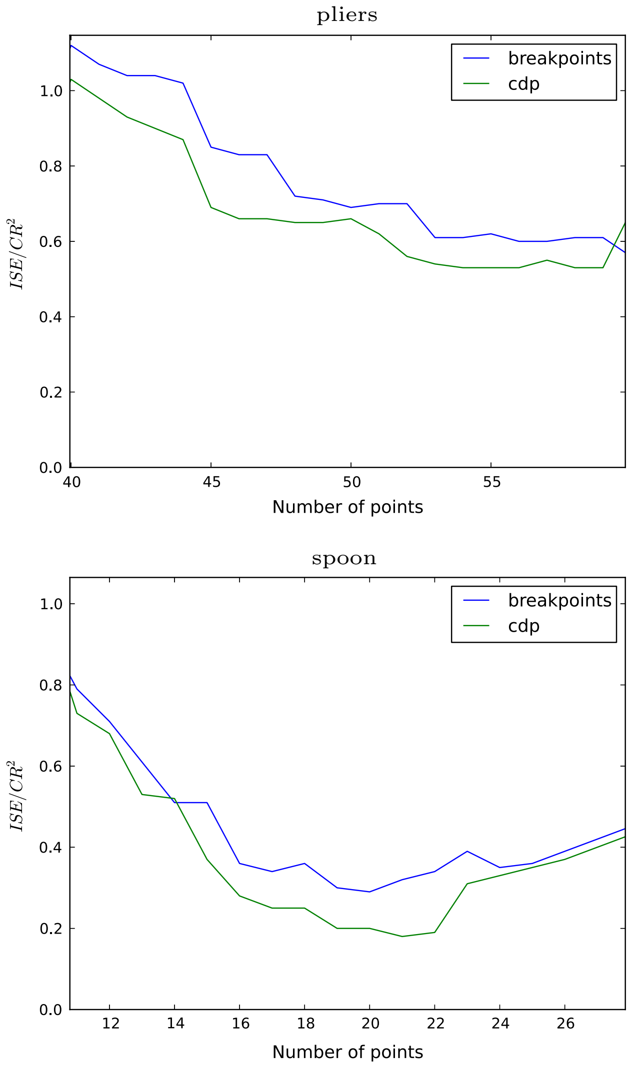


Fig. 7. Values of ISE/CR^2 obtained by the original Masood method (*breakpoints*) and the modified Masood method with our proposal (*CDP*), using pliers and spoon contours.

(f) Finally, Fig. 3(d) shows the set of *CDPs*. There are 29 *CDPs*, and they are contained in the 28 leaves of the extended concavity tree (see Fig. 4). Highlight that all *CDPs* are contained in the leaf nodes.

Fig. 5 shows the flowchart corresponding to the method to obtain the extended concavity tree and the set of *CDPs*.

The example of the chromosome contour shows that the three objectives of the proposed method have been fulfilled:

- The initial set of *CDPs* has been reduced because the number of candidates obtained by using our proposal is less than the number of breakpoints. This reduction is much higher in real contours.
- The set of *CDPs* obtained for the chromosome contour considers the different levels of detail of the different parts of the contour. For example, in Fig. 3(a), the first concave part is relatively straight, it is approximated by only one segment (1, 11) and only two points are added to set of *CDPs*. However, the other three concave parts need a high level of detail, and they are approximated by many segments. In this case, many points are added to the set of *CDPs*.

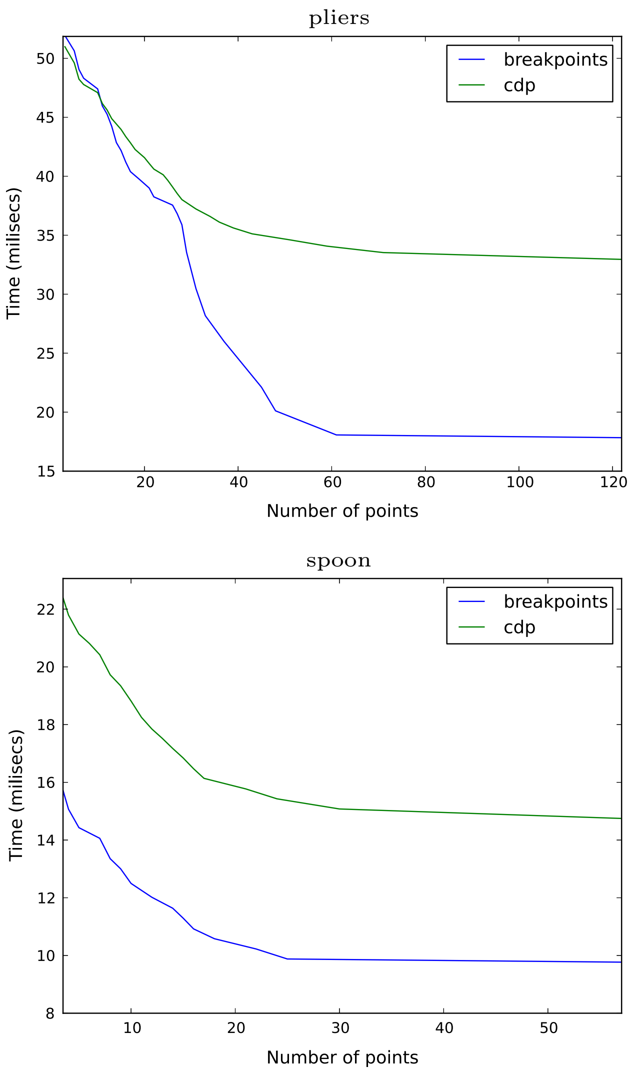


Fig. 8. Times obtained by the original Carmona method (*breakpoints*) and the modified Carmona method with our proposal (*CDP*), using pliers and spoon contours.

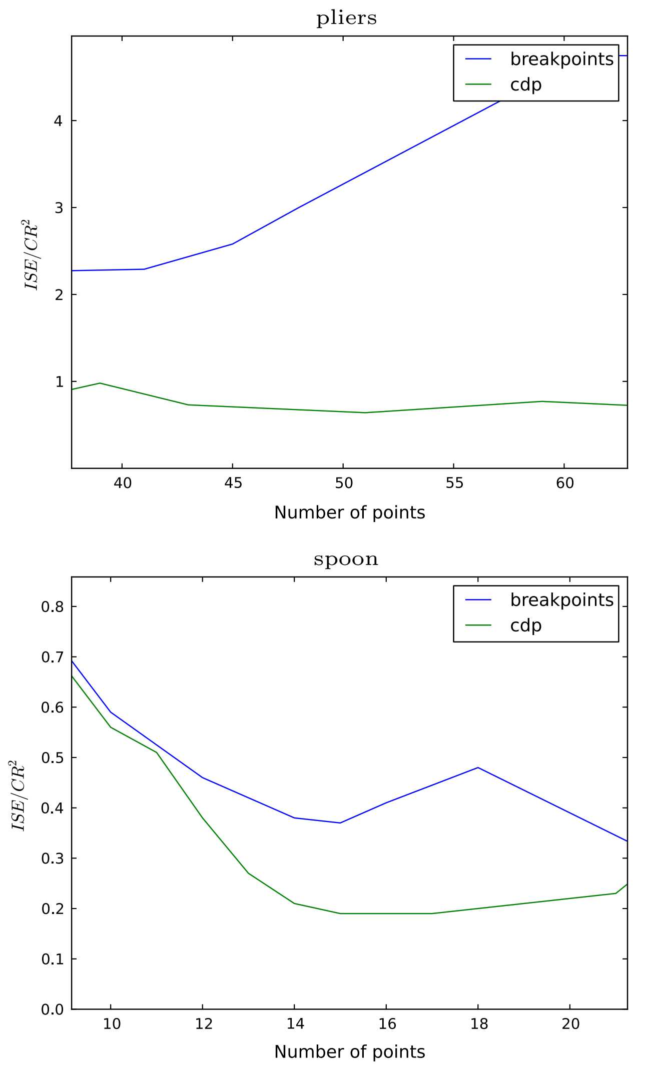


Fig. 9. Values of ISE/CR^2 obtained by the original Carmona method (*breakpoints*) and the modified Carmona method with our proposal (*CDP*), using pliers and spoon contours.

- All the relevant dominant points are added to set of *CDPs* because all the points of transition between parts of different level of detail have been taken into account. For example, the points 15, 29, 47 and 59 are added in the first level, the points 22, 37, 42 and 53 are added in the second level, and the points 16, 19, 25, 27, 49 and 51 are added in the third level.

Moreover, tests with real contours have shown that when the ISE/CR^2 and ISE/CR^3 measures have been used to calculate the error instead of ISE/CR, the number of nodes in the extended concavity tree is significantly decreased, and many relevant dominant points are discarded. Therefore, the discarded points never belonged to the initial set of *CDPs*, and the discarded points never belonged to the polygonal approximation. For this reason, the measurement ISE/CR has been selected.

In conclusion, the following features on the present proposal should highlighted:

- The present proposal considers the different levels of detail of the resultant subcontours of the concavity tree.
- The use of the concavity tree allows us to obtain dominant points of the contour as follows:
  - On the odd levels, relevant points of the convex parts of the contour or subcontours are added to the set of *CDPs*.
  - On the even levels, relevant points of the concave parts of the contour or subcontours are added to the set of *CDPs*.
- A non-parametric stop condition to obtain the set of *CDPs* is used. This condition depends on the local values of the measurement ISE/CR. Thus, this method considers the levels of detail of the different parts of the contour. In this way, many dominant points are added to the set of *CDPs* where the contour has a higher level of detail, and few dominant points are added to the set of *CDPs* where the contour is relatively straight.
- Redundant dominant points are selected, so no relevant dominant point will be discarded.
- We can assume that the time complexity is $O(N \log N)$ because the time complexity is defined by time complexity of convex hull calculation.

Table 1 contains the results obtained using real contours. This table shows the number of the initial set of *CDPs* (all the

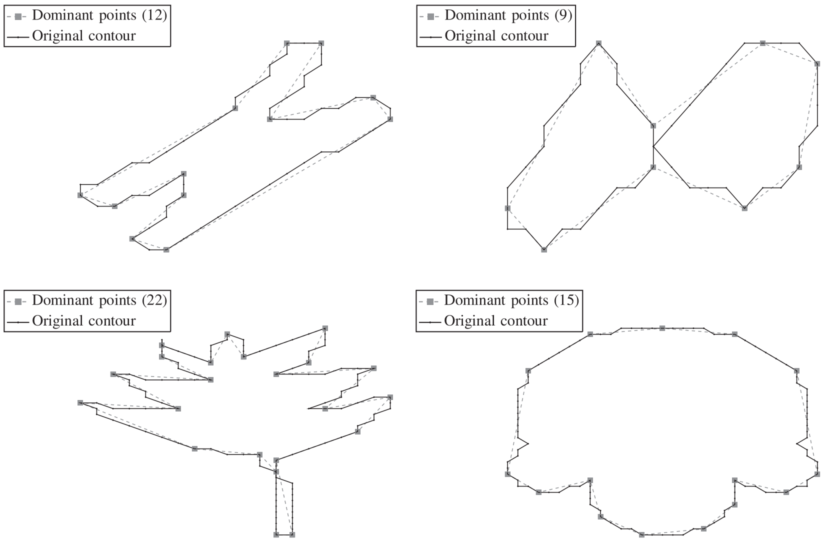


Fig. 10. Obtained polygonal approximations for synthetic contours.

Table 2
Comparison, using synthetic contours, with other methods.

Contour	*Method*	n_{dp}	*CR*	*ISE*	*Merit*
Chromosome ($N = 60$)	Teh and Chin [37]	15	4.00	7.20	61.79
	Wu [41]	16	3.75	4.70	77.64
	Marji and Siy [19]	10	6.00	10.01	87.58
	Carmona [6]	14	4.21	4.93	88.19
	Masood [20,21]	15	4.00	4.14	92.75
	Masood [22]	15	4.00	3.88	98.16
	Carmona [7]	15	4.00	4.27	90.57
	Parved [23]	10	6.00	14.34	66.19
	Proposed	12	5	5.82	100
Leaf ($N = 120$)	Teh and Chin [37]	29	4.14	14.96	50.13
	Wu [41]	24	5.00	15.93	64.50
	Marji and Siy [19]	17	7.06	28.67	82.45
	Carmona [6]	23	5.17	15.63	69.99
	Masood [20,21]	23	5.22	10.61	92.46
	Masood [22]	23	5.22	9.46	100
	Carmona [7]	23	5.22	10.68	92.03
	Parved [23]	21	5.71	13.82	89.99
	Proposed	22	5.45	11.42	95.28
Semicircles ($N = 102$)	Teh and Chin [37]	22	4.64	20.61	44.89
	Wu [41]	26	3.92	9.04	58.69
	Marji and Siy [19]	15	6.80	22.70	73.07
	Carmona [6]	24	4.21	9.88	65.54
	Masood [20,21]	26	3.92	4.91	88.37
	Masood [22]	26	3.92	4.05	100
	Carmona [7]	26	3.92	4.91	88.37
	Parved [23]	17	6.00	19.02	71.45
	Proposed	15	6.8	14.40	100
Infinity ($N = 45$)	Teh and Chin [37]	13	3.46	5.93	50.04
	Wu [41]	13	3.36	5.78	50.83
	Carmona [6]	10	4.40	5.56	68.69
	Masood [22]	11	4.09	2.90	99.05
	Carmona [7]	10	4.50	5.29	70.77
	Parved [23]	9	5.00	7.35	66.39
	Proposed	9	5	4.86	85.62

breakpoints) used in [20–22,7] and the number of the initial set of *CDPs* obtained with our proposal. Moreover, this Table shows the number of nodes of the extended concavity tree and its number of levels. The results can be summarised as follows:

- The set of *CDPs* is significantly reduced when the proposed method is used.

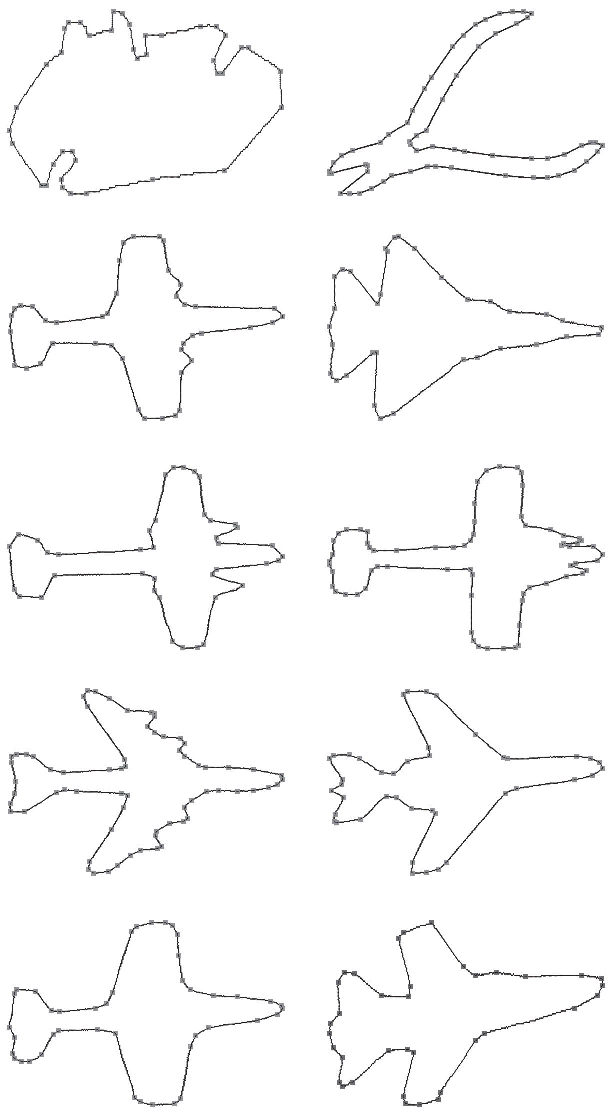


Fig. 11. Obtained polygonal approximations for real contours (I).

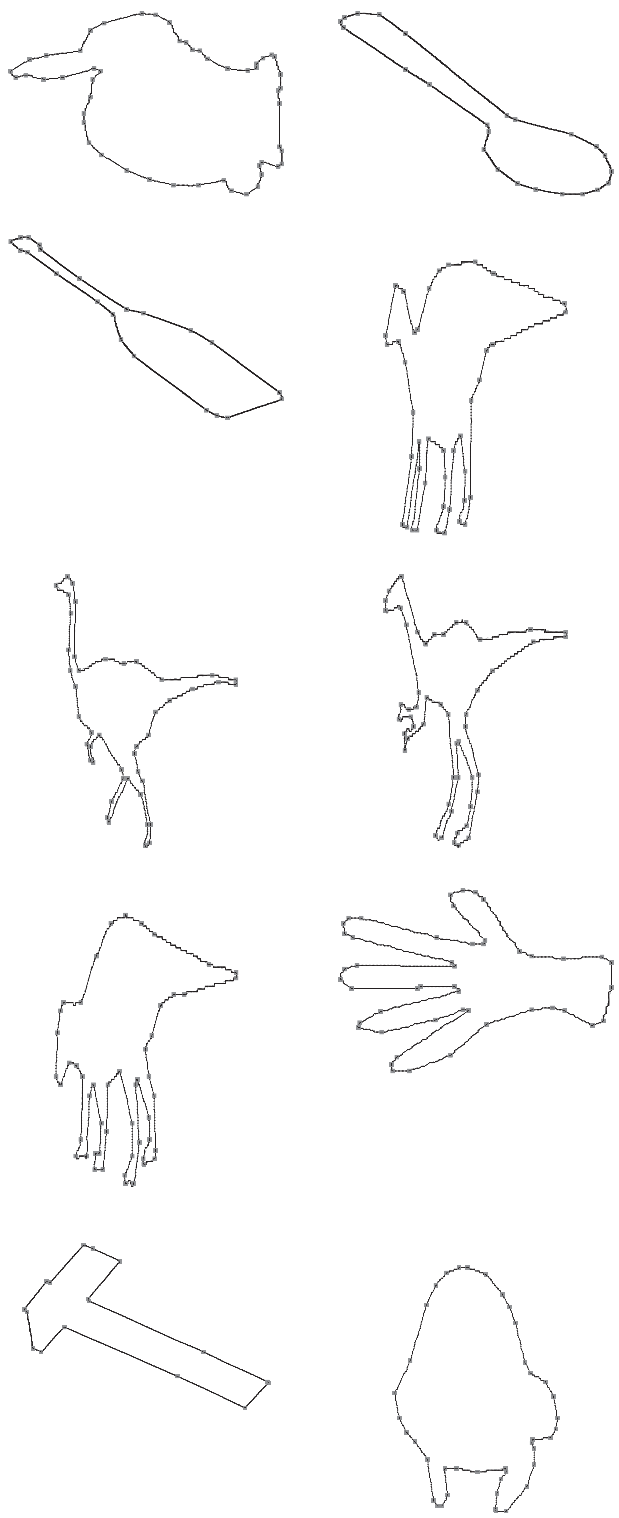


Fig. 12. Obtained polygonal approximations for real contours (II).

- The number of nodes is much lower than the number of breakpoints, and the number of levels of the tree is always less than or equal to 5.
- If the original contour has many small concave parts as a result of the digitization process, most of these parts do not appear in the extended concavity tree because the tree does not reach the necessary level of depth. For example, in spoon, screwdriver and hammer contours there are many small concave parts. In these cases, the set of *CDPs* obtained, and the number of levels in the extended concavity tree are even smaller than other real contours.

Obviously, the time spent in getting the breakpoints ($O(N)$) is less than the time spent in getting the set of *CDPs* in our proposal ($O(NlogN)$). However, the next experiments will demonstrate that the large reduction in the number of the set of *CDPs* obtained with our proposal compensates for the time.

5. Experiments, results and discussion

In this section, four experiments have been evaluated. The calculations were performed using a standard PC with Intel(R) Core i7(R) 8 CPU with 2.20 GHz, 5.7 Gb of RAM using Ubuntu(R) 10.04.

In the first and second experiments, the measurement ISE/CR^2 is used to compare the results instead of ISE/CR or ISE/CR^3 because the numerator and the denominator are balanced [7,19] and therefore a better balance between the error of the polygonal approximation and the number of points is achieved. This measurement will also be used in the third experiment as a stop condition to obtain polygonal approximations for all the contours used.

Figures shown in the first and second experiment are obtained in representative and real contours (pliers and spoon). The curves obtained in the other real contours are similar.

5.1. First experiment

In the first experiment, the set of *CDPs* obtained using the proposed method has been used as the initial set of *CDPs* in the Masood method [20,21] using real contours. Polygonal approximations with any number of points less than or equal to the number of *CDPs* has been obtained. The results have been compared to the results obtained when all breakpoints are used as the initial set of *CDPs* in this method.

Fig. 6 shows the times obtained for all the polygonal approximations with the number of points less than the number of *CDPs*. The results show that the times are vastly improved.

Fig. 7 shows the values of ISE/CR^2 obtained for different polygonal approximations with an appropriate number of points (balanced values of *ISE* and CR^2). The results show that the values of ISE/CR^2 are improved except in the spoon contour when the number of points of the polygonal approximation obtained is small (less than 14). However, for this reduced number of points, the polygonal approximations are very distorted, and they are very dissimilar to the original contour.

5.2. Second experiment

In this experiment, the set of *CDPs* obtained using the proposed method has been used as the initial set of *CDPs* in the Carmona method [7] using real contours. The results have been compared with the results obtained when all the breakpoints are used as the initial set of *CDPs* in this method.

Fig. 8 shows the times obtained for polygonal approximations with number of points less than the number of *CDPs*. The results show that the times obtained by the proposed method are worse than the times in the original method. However, the times obtained in the proposed method are less than 55 ms in all the real contours.

Fig. 9 shows the values of ISE/CR^2 obtained for different polygonal approximations with an appropriate number of points (balanced values of *ISE* and CR^2). The results show that the values of ISE/CR^2 are improved using our proposal.

Table 3
Summary of the results for real contours. N is the number of points of the contour, and M is the number of points of the polygonal approximation.

Contour	N	M	Optimised Masood method (breakpoints)			Optimised Masood method (proposal)		
			Efficiency	*Fidelity*	*Merit*	*Efficiency*	*Fidelity*	*Merit*
tin-openers	580	40	96.01	91.07	93.51	98.51	96.48	97.49
pliers	2040	62	97.41	96.38	96.90	97.99	97.17	97.58
plane1	1015	42	91.23	85.84	88.50	98.56	97.36	97.96
plane2	787	34	98.05	95.92	96.98	98.10	96.02	97.05
plane3	1073	39	97.96	94.12	96.02	97.99	94.21	96.08
plane4	1126	67	93.85	90.65	92.24	96.94	95.13	96.03
plane5	1098	67	93.02	89.47	91.23	93.29	89.84	91.55
plane6	921	42	99.23	97.97	98.60	98.10	95.16	96.62
plane7	897	41	97.28	93.45	95.34	98.22	95.61	96.91
plane8	722	36	88.48	81.37	84.85	96.54	93.57	95.04
rabbit	745	52	97.87	93.08	95.45	97.32	91.44	94.33
spoon	1370	24	97.75	94.43	96.07	97.75	94.43	96.07
screwdriver	1677	18	97.42	91.80	94.57	97.48	91.96	94.68
dinosaur1	795	42	96.51	93.94	95.22	97.14	94.98	96.05
dinosaur2	625	45	99.44	98.85	99.14	99.44	98.85	99.14
dinosaur3	889	52	97.73	94.79	96.25	96.84	92.90	94.85
dinosaur4	779	51	97.04	95.04	96.21	97.53	95.28	96.40
hand	1041	51	99.72	99.53	99.63	99.72	99.53	99.63
hammer	1583	17	98.05	94.80	96.41	97.98	94.62	96.29
turtle	553	46	86.22	85.31	85.76	97.49	96.96	97.23
Average			96.01	92.89	94.44	97.65	95.08	96.35

Table 4
Comparison, using synthetic contours, with the Roussillon method [31].

Contour	Method	M	CR	ISE	Merit	Length
Chromosome ($N = 95$)	Roussillon [31]	27	3.52	21.09	43.31	64.43
	Proposed[a]	25	3.8	20.70	47.93	77.39
Leaf ($N = 161$)	Roussillon [31]	38	4.24	32.33	45.34	111.08
	Proposed	38	4.24	14.81	88.97	127.63
Semicircles ($N = 137$)	Roussillon [31]	34	4.03	31.20	35.79	102.97
	Proposed	34	4.03	19.22	59.14	114.51
Infinity ($N = 67$)	Roussillon [31]	18	3.72	13.22	46.18	46.17
	Proposed	18	3.72	7.27	82.55	54.16

[a] In Chromosome contour, the maximum number of points of the polygonal approximation obtained using our proposal is 25.

5.3. Third experiment

In this experiment, our proposal has been used to obtain polygonal approximations in some contours. For this purpose, the following conditions have been applied:

- The optimised Masood method [22] has been used with the initial set of *CDPs* obtained by our proposal.
- The minimum value of the measurement ISE/CR^2 has been used as the stop condition [8].

Fig. 10 shows the polygonal approximations obtained for the synthetic contours using the proposed method.

Table 2 shows the results obtained, using synthetic contours, compared with other methods. The merit measurement of Rosin [30] has been used to compare. The results of the proposed method are better than other methods, except the Masood method [22] (leaf and infinity contours). For this reason, this method will be compared with our proposal using real contours. On the other hand, these synthetic contours are very small with a reduced number of points and the results should be confirmed using real contours. Figs. 11 and 12 show the polygonal approximations obtained for the real contours using the proposed method.

The polygonal approximations obtained using real contours have been compared with the polygonal approximations obtained by the optimised Masood method [22] when all breakpoints are used as the set of *CDPs*. For this purpose, the merit measurement of Rosin [30] has been used to compare. The results are shown in Table 3.

Table 5
Comparison, using real contours, with the Roussillon method [31] using polygonal approximations with the same number of points.

Contour	Method	M	CR	ISE	Merit	Length
Tin-openers ($N = 781$)	Roussillon [31]	107	7.30	125.99	45.32	621.19
	Proposed	107	7.30	125.32	46.05	652.78
Rabbit ($N = 1003$)	Roussillon [31]	150	6.69	168.30	61.81	800.48
	Proposed	150	6.69	98.72	92.49	851.59
Dino1 ($N = 1041$)	Roussillon [31]	146	7.13	168.48	49.90	841.91
	Proposed	146	7.13	234.54	27.85	895.75
Dino2 ($N = 833$)	Roussillon [31]	138	6.04	154.49	51.97	663.91
	Proposed	138	6.04	198.66	38.80	713.87
Dino4 ($N = 1023$)	Roussillon [31]	159	6.43	167.56	55.30	823.26
	Proposed	159	6.43	162.77	56.98	882.01
turtle ($N = 781$)	Roussillon [31]	106	7.37	137.15	56.92	608.04
	Proposed	106	7.37	94.03	85.55	693.86

Table 6
Comparison, using real contours, with the Roussillon method [31] using the optimised polygonal approximations obtained with our proposal.

Contour	Method	M	CR	ISE	Merit	Length
tin-openers ($N = 781$)	Roussillon [31]	107	7.30	125.99	45.42	621.19
	Proposed	36	21.69	142.75	96.39	632.73
pliers ($N = 2855$)	Roussillon [31]	293	9.74	454.76	62.86	2239.14
	Proposed	54	52.87	620.42	97.78	2244.71
plane1 ($N = 1353$)	Roussillon [31]	230	5.88	217.49	53.01	1064.11
	Proposed	41	33	335.77	92.67	1072.57
plane2 ($N = 1177$)	Roussillon [31]	177	6.65	188.44	55.92	850.95
	Proposed	31	37.97	260.14	98.48	856.01
plane3 ($N = 1447$)	Roussillon [31]	266	5.44	238.14	64.59	1121.73
	Proposed	35	41.34	608.39	98.90	1129.62
plane4 ($N = 1451$)	Roussillon [31]	250	5.80	256.33	37.30	1166.11
	Proposed	55	26.38	316.72	94.63	1179.12
plane5 ($N = 1565$)	Roussillon [31]	252	6.21	278.92	55.84	1191.43
	Proposed	52	30.10	396.74	97.87	1201.46
plane6 ($N = 1341$)	Roussillon [31]	186	7.21	229.92	60.60	1020.56
	Proposed	41	32.71	323.93	95.64	1030.13
plane7 ($N = 1191$)	Roussillon [31]	188	6.34	191.74	57.58	937.46
	Proposed	37	32.19	297.93	94.87	945.11
plane8 ($N = 1003$)	Roussillon [31]	155	6.47	184.14	40.82	775.65
	Proposed	35	28.66	215.72	95.49	784.41
rabbit ($N = 1003$)	Roussillon [31]	150	6.69	168.30	61.81	800.48
	Proposed	36	27.86	347.23	98.67	803.82
dino1 ($N = 1041$)	Roussillon [31]	146	7.13	168.48	49.90	841.91
	Proposed	39	26.69	286.27	80.79	857.16
dino2 ($N = 833$)	Roussillon [31]	138	6.04	154.49	51.97	663.91
	Proposed	35	23.08	269.18	98.03	661.75
dino3 ($N = 1257$)	Roussillon [31]	192	6.55	216.04	52.79	1010.79
	Proposed	48	26.19	351.41	94.29	1022.02
dino4 ($N = 1023$)	Roussillon [31]	159	6.43	167.56	55.30	823.26
	Proposed	41	24.95	298.92	93.08	836.15
hand ($N = 1413$)	Roussillon [31]	259	5.46	270.53	51.64	1124.22
	Proposed	38	37.18	617.35	96.00	1127.51
hammer ($N = 2501$)	Roussillon [31]	416	6.01	421.07	76.21	1836.12
	Proposed	15	166.73	1064.78	98.85	1837.75
turtle ($N = 781$)	Roussillon [31]	106	7.37	137.15	56.92	608.04
	Proposed	30	26.03	205.26	92.79	611.82

From this table, the following conclusions can be reached:

- For the present proposal, merit measurement is greater than 94% (except plane5) in all tested contours.
- In most contours (except plane6, rabbit, dinosaur3 and hammer for which the results are slightly worse), our proposal obtains better results.
- The mean values of efficiency, fidelity and merit are better for the present proposal.

The non-parametric Wilcoxon signed-rank test has been used to compare the values of merit. The results show that the values of merit when our proposal is used are better than the values obtained with breakpoints using a high significance level ($\alpha = 0.05\%$). Moreover, the mean value of merit obtained by our proposal is 96.35 while the mean value obtained with all the breakpoints is 94.44.

Considering the polygonal approximations obtained using the present proposal in the optimised Masood method [22] with the measurement ISE/CR^2, we can conclude the following:

- The polygonal approximations obtained are very similar to the original contours.
- The polygonal approximations have an appropriate number of points (neither excessive nor reduced), and they fit properly into the original contour.
- The number of dominant points in different parts of a contour depends on the level of detail in these parts. Straight lines produce very few dominant points, and the curved lines produce many dominant points.
- The merit values for tested real contours are very high.

5.4. Fourth experiment

In this experiment our proposal has been compared with the Roussillon method [31]. This method obtains minimal length polygons (*MLP*), and these polygons can be used as polygonal approximations without changing important geometrical and topological properties. For this purpose synthetic and real contours has been used and the following conditions have been applied:

- Since the Roussillon method uses 4-connectivity, the contours have been transformed from 8-connectivity to 4-connectivity.
- The optimised Masood method [22] has been used with the initial set of *CDPs* obtained by our proposal.
- The merit measurement of Rosin [30] has been used to compare the results.
- In order to compare the results, the polygonal approximation obtained with our proposal has the same number of points as the *MLP* obtained using the Roussillon method (Tables 4 and 5).
- In real contours, the best polygonal approximations obtained using our proposal with the optimised Masood method [22] have been compared to the polygonal approximations obtained using the Rousillon method (Table 6).

Fig. 13 shows the polygonal approximations obtained in synthetic contours using the Roussillon method. Black lines depict the original contour, blue grids depict the inner and the outer polygon and red lines depict the *MLP*. Fig. 14 shows the polygonal approximations using our proposal with 4-connectivity.

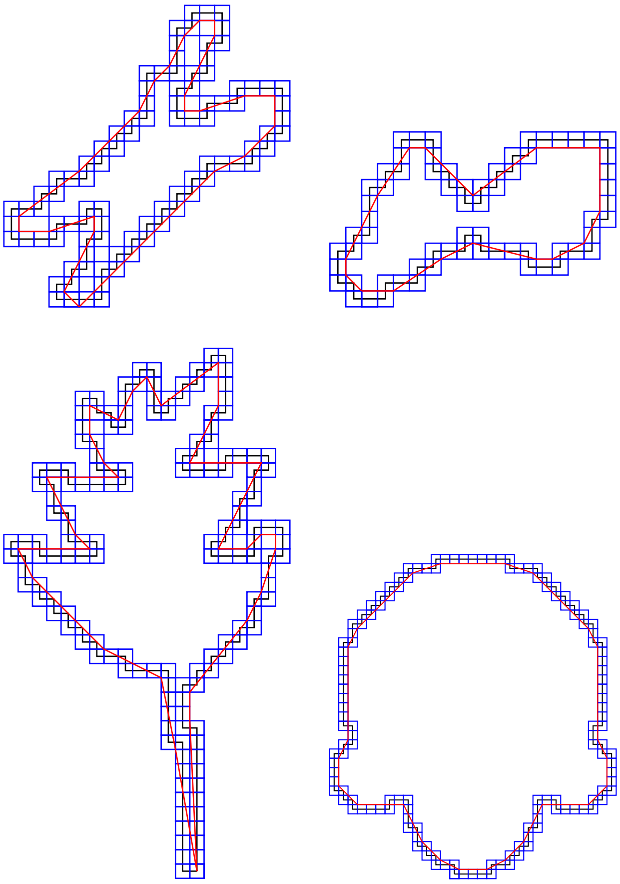


Fig. 13. Obtained polygonal approximations using the Roussillon method [31]. Black lines depict the original contour, blue grids depict the inner and the outer polygon and red lines depict the *MLP*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4 shows the results obtained using synthetic contours, compared with the Roussillon method [31]. This table shows that the results of *Merit* for the proposed method are better than the Roussillon method for all the synthetic contours, however the values of length are better in the Rousillon method. These results are due to our proposal optimises the error of the polygonal approximation while the Rousillon method optimises the length of the polygonal approximation.

Table 5 shows the results obtained using real contours, compared with the Rousillon method [31], when the number of points of the polygonal approximations are equal. In this case six contours are shown. In the remaining contours the number of *CDPs* is less than the number of points of the polygonal approximations obtained with the MLP method. This table shows that the results of *Merit* for the proposed method are better in four contours, and the Rousillon method is better in two contours.

Table 6 shows the results obtained using real contours, compared with the Rousillon method [31]. In this case the best polygonal approximations obtained using our proposal with the optimised Masood method [22] have been compared. This table shows that the results of *Merit* for the proposed method are better than the Rousillon method for all the real contours, however the values of length are slightly better in Rousillon method.

6. Conclusions

An efficient method to improve merge methods to obtain polygonal approximations in closed 2D contours is presented. The new proposal is non-parametric and relies on the split approach.

The extended concavity tree is used to obtain the initial set of *CDPs*. The new proposal considers the level of detail of the different parts of the original contour. For this reason, a non-parametric stop

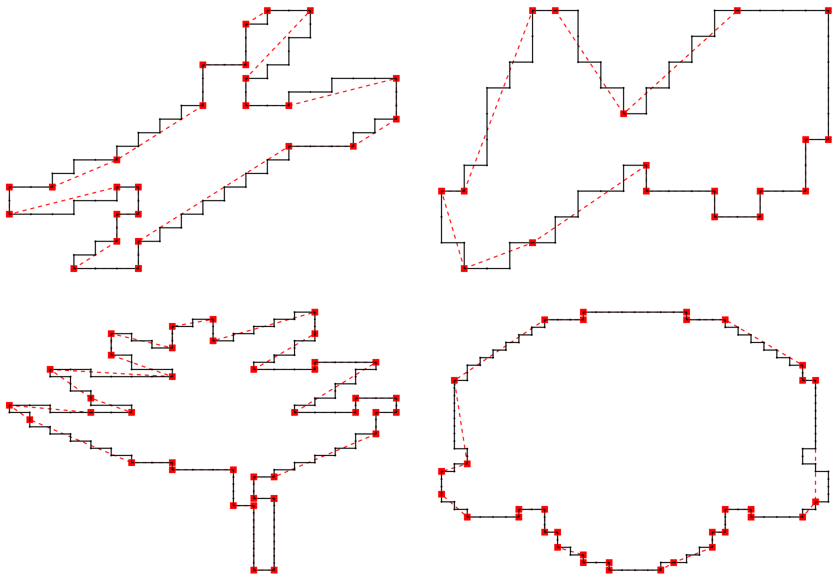


Fig. 14. Obtained polygonal approximations using our proposal with 4-connectivity.

condition, relying on the measurement *ISE*/*CR*, is used to obtain the set of *CDPs*. More relevant information about the overall contour is provided, and the different levels of detail are considered. Therefore, few dominant points are added to the set of *CDPs* where the contour is relatively straight or straight, and many dominant points are added to *CDP* where the contour has a higher level of detail (contour is curved).

When the proposed method is used to obtain the initial set of *CDPs*, and this set is used in the Masood method [20,21], and the Carmona method [7], the value of the measurement ISE/CR^2 is improved in all cases. The times obtained are also improved, except in the Carmona method, where the times are slightly worse.

When the proposed method has been used in real contours, and it is applied in the optimised Masood method [22], the merit values have been greater than 91% in all tested contours and the merit values improve the results obtained by the optimised Masood method [22] when all breakpoints are used as the initial set of *CDPs*.

References

[1] F. Attneave, Some informational aspects of visual perception, Phychol. Rev. 61 (1954) 189–193.
[2] R. Bellman, On the approximation of curves by line segments using dynamic programming, Commun. ACM 4 (6) (1961) 28.
[3] B. Batchelor, Hierarchical shape description based upon convex hulls of concavities, J. Cybernet. 10 (1980) 205–210.
[4] B. Batchelor, Shape descriptors for labeling concavity trees, J. Cybernet. 10 (1980) 233–237.
[5] G. Borgefors, G. Sanniti di Baja, Analyzing nonconvex 2D and 3D patterns, Comput. Vision Image Understand. 63 (1) (1996) 145–157.
[6] 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.
[7] A. Carmona-Poyato, F.J. Madrid-Cuevas, R. Medina-Carnicer, R. Munoz-Salinas, Polygonal approximation on digital planar curves through breakpoint suppression, Pattern Recognit. 43 (2010) 14–25.
[8] A. Carmona-Poyato, R. Medina-Carnicer, R. Munoz-Salinas, E. Yeguas-Bolivar, On stop conditions about methods to obtain polygonal approximations relied on breakpoint suppression, Image Vision Comput. 30 (2012) 513–523.
[9] D.H. Douglas, T.K. Peucker, Algorithm for the reduction of the number of points required to represent a line or its caricature, Can. Cartographer 10 (1973) 112–122.
[10] J.G. Dunham, Optimum uniform piecewise linear approximation of planar curves, IEEE Trans. Pattern Anal. Mach. Intell. 8 (1986) 67–75.
[11] O. El Badawy, M.S. Kamel, Hierarchical representation of 2-D shapes using convex polygons: a contour-based approach, Pattern Recognit. Lett. 26 (2005) 865–877.
[12] D.S. Guru, R. Dinesh, P. Nagabhushan, Boundary based corner detection and localization using new cornerity index: a robust approach, in: 1st Canadian Conference on Computer and Robot Vision, 2004, pp. 417–423.
[13] S.C. Huang, Y.N. Sung, Polygonal approximation of digital using genetic algorithms, Pattern Recognit. 32 (1999) 1409–1420.
[14] S. Hsin-Teng, H. Wu-Chih, A rotationally invariant two-phase scheme for corner detection, Pattern Recognit. 29 (1996) 819–828.
[15] H. Imai, M. Iri, Polygonal approximations of a curve (Formulations and algorithms), in: G.T. Toussaint (Ed.), Computational Morphology, North-Holland, Amsterdam, 1988, pp. 71–86.
[16] G. Klette, Recursive computation of minimum-length polygons, Comput. Vision Image Understand. 117 (2013) 386–392.
[17] L.J. Latecki, R. Lakamper, Convexity rule for shape decomposition based on discrete contour, Evol. Comput. Vision Image Understand. 73 (1999) 441–454.
[18] D.G. Lowe, Three-dimensional object recognition from single two dimensional images, Artif. Intell. 31 (1987) 355–395.
[19] M. Marji, P. Siy, Polygonal representation of digital planar curves through dominant point detection – a nonparametric algorithm, Pattern Recognit. 37 (2004) 2113–2130.
[20] A. Masood, S.A. Haq, A novel approach to polygonal approximation of digital curves, J. Visual Commun. Image Represent. 18 (2007) 264–274.
[21] A. Masood, Dominant point deletion by reverse polygonization of digital curves, Image Vision Comput. 26 (2008) 702–715.
[22] A. Masood, Optimized polygonal approximation by dominant point deletion, Pattern Recognit. 41 (2008) 227–239.
[23] M.T. Parvez, S.A. Mahmoud, Polygonal approximation of digital planar curves through adaptive optimizations, Pattern Recognit. Lett. 31 (2010) 1997–2005.
[24] T. Pavlidis, S.L. Horowitz, Segmentation of plane curves, IEEE Trans. Comput. 23 (8) (1974) 860–870.
[25] J.C. Perez, E. Vidal, Optimum polygonal approximation of digitized curves, Pattern Recognit. Lett. 15 (1994) 743–750.
[26] A. Pikaz, I. Dinstein, An algorithm for polygonal approximation based on iterative point elimination, Pattern Recognit. Lett. 16 (1995) 557–563.
[27] A. Pikaz, I. Dinstein, Optimal polygonal approximation of digital curves, Pattern Recognit. 28 (1995) 373–379.
[28] U. Ramer, An iterative procedure for the polygonal approximation of plane curves, Comput. Graph. Image Process. 1 (1972) 244–256.
[29] B.K. Ray, K.S. Ray, A non-parametric sequential method for polygonal approximation of digital curves, Pattern Recognit. Lett. 15 (1994) 161–167.
[30] P.L. Rosin, Techniques for assessing polygonal approximation of curves, IEEE Trans. Pattern Anal. Mach. Intell. 19 (6) (1997) 659–666.
[31] T. Roussillon, I. Sivignon, Faithful polygonal representation of the convex and concave parts of a digital curve, Pattern Recognit. 44 (2011) 2693–2700.
[32] M. Salotti, An efficient algorithm for the optimal polygonal approximation of digitized curves, Pattern Recognit. Lett. 22 (2001) 215–221.
[33] M. Salotti, Optimal polygonal approximation of digitized curves using the sum of square deviations criterion, Pattern Recognit. 35 (2002) 435–443.
[34] D. Sarkar, A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves, Pattern Recognit. Lett. 14 (1993) 959–964.
[35] J. Sklansky, Measuring concavity on a rectangular mosaic, IEEE Trans. Comput. C-21 (1972) 1355–1364.
[36] J. Sklansky, J. Gonzales, Fast polygonal approximation of digitized curves, Pattern Recognit. 12 (1980) 327–331.
[37] C.H. Teh, R.T. Chin, On the detection of dominant points on digital curves, IEEE Trans. Pattern Anal. Mach. Intell. 11 (1989) 859–872.
[38] B. Wang, H.Z. Shu, B.S. Li, Z.M. Niu, A mutation-particle swarm algorithm for error-bounded polygonal approximation of digital curves, in: Proc. Int. Conf. Intelligent Computing: Advanced Intelligent Computing Theories and Applications with Aspects of Theoretical and Methodological Issues. LNCS, 2662, 2008, pp. 1149–1155.
[39] B. Wang, H.Z. Shu, L.M. Luo, A genetic algorithm with chromosome repairing for min-# and min-e polygonal approximation of digital curves, J. Visual Commun. Image Represent. 20 (1) (2009) 45–56. 2009.
[40] J. Wang, Z. Kuang, X. Xu, Y. Zhou, Discrete particle swarm optimization based on estimation of distribution for polygonal approximation problems, Exp. Syst. Appl. 36 (2009) 9398–9408.
[41] W.Y. Wu, An adaptive method for detecting dominant points, Pattern Recognit. 36 (2003) 2231–2237.
[42] Y. Xiao, J.J. Zhou, H. Yan, An adaptive split-and-merge method for binary image contour data compression, Pattern Recognit. Lett. 22 (2001) 299–307.
[43] J. Xu, Hierarchical representation of 2-D shapes using convex polygons: a morphological approach, Pattern Recognit. Lett. 18 (1997) 1009–1017.
[44] A tabu search approach to polygonal approximation of digital curves. Int. J. Pattern Recognit. Artif. Intell. 13 (1999) 1061–1082.
[45] P.Y. Yin, Polygonal approximation of digital curves using the state-of-the-art metaheuristics, in: Obinata, G., Dutta, A. (Eds.), Vision Systems: Segmentation and Pattern Recognition, 2007, pp. 451–464.

Chapter 3

Second contribution: Novel method to obtain the optimal polygonal approximation of digital planar curves based on Mixed Integer Programming

J. Vis. Commun. Image R. 30 (2015) 106–116

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Novel method to obtain the optimal polygonal approximation of digital planar curves based on Mixed Integer Programming ☆

E.J. Aguilera-Aguilera *, A. Carmona-Poyato, F.J. Madrid-Cuevas, R. Muñoz-Salinas

Department of Computing and Numerical Analysis, Córdoba University, 14071-Córdoba, Spain

ARTICLE INFO

Article history:
Received 27 November 2014
Accepted 22 March 2015
Available online 30 March 2015

Keywords:
Polygonal approximation
Digital planar curve
Mixed Integer Programming
Discrete optimization
Dominant points
Breakpoints
Integral square error
Optimization

ABSTRACT

Polygonal approximations of digital planar curves are very useful for a considerable number of applications in computer vision. A great interest in this area has generated a huge number of methods for obtaining polygonal approximations. A good measure to compare these methods is known as Rosin's merit. This measure uses the optimal polygonal approximation, but the state-of-the-art methods require a tremendous computation time for obtaining this optimal solution.

We focus on the problem of obtaining the optimal polygonal approximation of a digital planar curve. Given N ordered points on a Euclidean plane, an efficient method to obtain M points that defines a polygonal approximation with the minimum distortion is proposed.

The new solution relies on Mixed Integer Programming techniques in order to obtain the minimum value of distortion. Results, show that computation time for the new method dramatically decreases in comparison with state-of-the-art methods for obtaining the optimal polygonal approximation.

1. Introduction

Polygonal approximations of digital planar curves are important for a great number of applications, for example image analysis [14], shape analysis [9] and digital cartography [7]. For this reason, a great interest in this area has generated a huge number of methods for obtaining polygonal approximations [22,4,3].

The polygonal approximation problem can be formulated as two different optimization problems [12]:

- min-#: Given a distortion threshold obtain the polygonal approximation with the minimum number of points M.
- min-ε: Minimize the distortion error for a polygonal approximation given the number of points M of the approximation.

Methods in the literature can be classified into suboptimal and optimal, depending on the type of the solution obtained, that is, optimal algorithms guarantee optimal polygonal approximations, whereas, suboptimal methods do not assure the optimality of the solution obtained.

Usually, optimal methods to obtain polygonal approximations are used to compare suboptimal methods. This is done by using the error associated to the optimal polygonal approximation.

The associated error measures the distortion introduced by the polygonal approximation regarding the original digital planar curve. The distortion decreases as the number of points of the polygonal approximation increase, but many of these points may be redundant, that is, they do not provide relevant information. Therefore, a good balance between the distortion and the number of points is a desirable characteristic for polygonal approximations. Due to these two objectives are opposite, achieving a good balance between them is complicated.

A widely used measure of the distortion associated to the polygonal approximation is known as Integral Square Error (ISE). Let's suppose that a curve S has been approximated using a segment composed of points s_i and s_j, then, a distortion measure $\Delta(i,j)$ can be defined as follows:

$$\Delta(i,j) = \sum_{k=i}^{j} d(s_k, \overline{s_i s_j})^2 \tag{1}$$

where the term $d(s_k, \overline{s_i s_j})$ indicates the orthogonal distance from the point s_k to the segment $\overline{s_i s_j}$. Fig. 1 shows a curve that is approximated using a segment $\overline{s_i s_j}$, therefore the distortion $\Delta(i,j)$ is the sum of the squared orthogonal distances from points between point i and j to segment $\overline{s_i s_j}$, that is, $\Delta(i,j) = d_1^2 + d_2^2 + d_3^2$. The distortion

☆ This paper has been recommended for acceptance by Yehoshua Zeevi.

* Corresponding author. Fax: +34 957218630.
E-mail addresses: i22agage@uco.es (E.J. Aguilera-Aguilera), ma1capoa@uco.es (A. Carmona-Poyato), ma1macuf@uco.es (F.J. Madrid-Cuevas), in1musar@uco.es (R. Muñoz-Salinas).

http://dx.doi.org/10.1016/j.jvcir.2015.03.007
1047-3203/

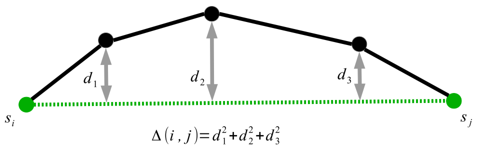


Fig. 1. Distortion sum of an approximated line segment.

measure ISE associated to a polygonal approximation with M points can be defined as follows:

$$\mathrm{ISE} = \sum_{k=0}^{M-1} \Delta(i_k, j_k) \tag{2}$$

where $i_k = j_{k-1}$ (modular arithmetic is assumed). Thus, the distortion associated to each of the M segments of the polygonal approximation are added. The original curve has associated a value of ISE equal to zero. The minimum number of points, that form a polygonal approximation with a value of distortion of zero, are named as breakpoints in the literature.

To compare the different methods to obtain polygonal approximations, Rosin [18] introduced two components named as fidelity and efficiency. Fidelity defines how well the polygonal approximation fits the optimal polygonal approximation in terms of the approximation error. Efficiency measures how compact is the polygonal approximation supplied regarding the optimal polygon with the same error.

Fidelity is defined as:

$$\text{Fidelity} = \frac{\mathrm{E_{opt}}}{\mathrm{E_{approx}}} \times 100 \tag{3}$$

where $\mathrm{E_{approx}}$ is the distortion (ISE) of the polygonal approximation and $\mathrm{E_{opt}}$ is the distortion of the optimal polygonal approximation. Both distortion values, $\mathrm{E_{approx}}$ and $\mathrm{E_{opt}}$, are obtained for the same number of points.

Efficiency is defined as:

$$\text{Efficiency} = \frac{\mathrm{M_{opt}}}{\mathrm{M_{approx}}} \times 100 \tag{4}$$

where $\mathrm{M_{approx}}$ is the number of segments of the polygonal approximation and $\mathrm{M_{opt}}$ is the number of segments that an optimal polygonal approximation would require to obtain the same error.

These two measures may vary depending on the curve. In order to avoid this problem Rosin [18] used a geometric mean of these two measures named as merit. The Rosin's merit is defined as:

$$\text{Merit} = \sqrt{\text{Fidelity} \times \text{Efficiency}} \tag{5}$$

The advantage of using Rosin's merit as a measure for comparing algorithms to obtain polygonal approximations is, that these polygonal approximations may have a different number of points. Therefore, all algorithms can be evaluated in a fair way.

To compute Rosin's merit, the optimal polygonal approximation is required. For this purpose, the method proposed by Perez and Vidal [17] is used. This algorithm relies on the Dynamic Programming (DP) technique, to obtain the optimal polygonal approximation with a fixed number of points M for a digitized planar curve with N points. The main drawback of this method is the complexity, $O(MN^2)$ for closed curves. This complexity also increases because the initial point of the polygonal approximation has to be set as a parameter to the method [17]. Therefore, to obtain the optimal polygonal approximation, the method must try all points as initial point. Taking into account this problem, final complexity for obtaining the optimal polygonal approximation for a closed curve is $O(MN^3)$.

Some authors has proposed great improvements over the Dynamic Programming method for reducing the computation time. Horng and Li [11] 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 algorithm does not assure that the solution obtained is optimal.

Another attempt to reduce the computation time was introduced by Salotti [20]. This method used the A* algorithm to search in a graph formed by the points of the curve. This graph has a root node, which is the starting point of the curve. Therefore, this solution needs to try all points as initial point (initial node in the graph) to obtain the optimal solution. Nevertheless, this solution has a complexity close to $O(N^2)$[20], where N is the number of points of the curve.

Masood [15] proposed another framework of optimization. This method selects an initial set of points and deletes one point per iteration depending on the error associated to this point. After removing the point a local optimization process search the optimal position of the remaining points that minimizes the distortion. This process does not guarantee that the solution obtained is optimal.

In this paper a new method to optimally solve the min-ε problem is proposed. The new method relies on Mixed Integer Programming (MIP) technique and has some advantages over previous algorithms: (1) no initial point is needed to be set as a parameter, (2) time required to compute optimal solution is significantly lower than the state-of-the-art alternatives and (3) and the solution obtained is optimal.

The rest of this paper is structured as follows: Section 2 describes the proposed method. Section 3 describes the experiments carried out and results obtained by the proposed method. Section 4 discusses some relevant aspects of the proposed method and experimental results. Finally, Section 5 shows the main conclusions.

2. MIP model formulation

The problem of obtaining the optimal polygonal approximation of a planar curve has been solved as an optimization problem, using mainly dynamic programming techniques. We propose to state the polygonal approximation problem as a Mixed Integer Programming problem (MIP).

A MIP problem is a mathematical problem in which an objective function has to be minimized or maximized and is subject to a set of linear constraints. MIP problems may contain a subset of the variables that has also the constraint of being integer.

The problem formulation has an objective function defined as:

$$z = \min c^T x,\ z, x \in \mathbb{R}^n \tag{6}$$

which has to be a linear expression formed by a vector x of decision variables and a cost vector c. This objective function is subject to a set of constraints defined as:

$$Ax \leqslant b \tag{7}$$

where A is called constraint matrix. Decision variables may take values between an upper and a lower bound which is defined as:

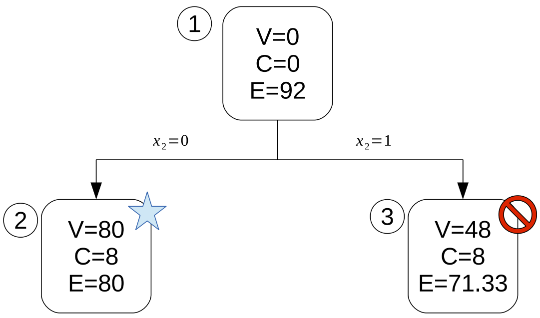
$$l \leqslant x \leqslant u \tag{8}$$

Some decision variables are required to take integer values. Integer variables that must take values 0 or 1 are called binary variables and play a special role in MIP modeling and solving.

Solving MIP usually includes two different stages. First, the problem is solved with a relaxation of the integer constraint, that is, the problem is solved by using the Simplex method (introduced by Dantzig [6]) as if there were no integer restrictions. This process

$$z = max\ 45x_1 + 48x_2 + 35x_3$$
$$s.t.\quad 5x_1 + 8x_2 + 3x_3 \leqslant 10$$
$$x_1, x_2, x_3 \in \{0,1\}$$

(a) MIP model



(b) Tree generation

Fig. 2. Branch & Bound example. The model Fig. 2(a) defines a linear objective function to be optimized subject to a set of constraints. The algorithms generates a tree, Fig. 2(b), where the method estimates an upper bound of the objective function (value E). Each leaf node represents a solution with an objective function value V associated for some fixed integer values of the decision variables, which satisfies the model constraints ($C = 5x_1 + 8x_2 + 3x_3$).

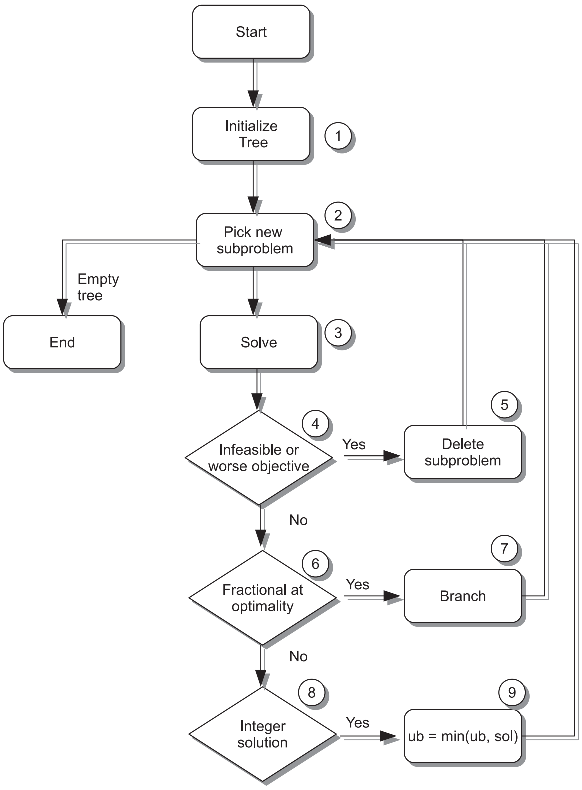


Fig. 3. Branch & Bound algorithm.

obtain a value of the objective function that is a lower bound on the MIP since we are minimizing over a large set that encompasses all the integer solutions.

The next phase uses the algorithm named as Branch & Bound. This method was introduced by Land and Doig [13] and relies on a process of searching in a tree. This algorithm works by enumerating the possible combinations of the integer variables in a tree. Each node of the tree is a continuous optimization problem, based on the original problem, but with some decision variables set to a fixed value. The remaining variables are allowed to take some value of the possible integer range provided. The root of this tree is the original problem with all of the integrality constraint relaxed. The algorithm generates the tree by selecting an integer variable x_i and add a node for each integer value of the possible range. Then, the method selects a node of the tree, and solves the problem with the variable x_i fixed to a value and the constraints updated. Four possible results can be obtained:

- The subproblem is infeasible if the Simplex obtained is unbounded, then, any further restriction of the subproblem would also be infeasible. Therefore, the current node of the tree could be pruned.
- The subproblem is feasible, but the objective function is worse than a previous known integer solution, then, no children of the current subproblem could get a better solution. This node could be pruned.
- All the integer constraints are satisfied for the current subproblem, and the objective function is better than the best previous known solution, so, this solution and objective function are recorded as the best feasible solution found. This node is a leaf of the tree and no children nodes can be generated.
- If none of the above occurs, then, one variable x_i is fractional (not integer) at optimality, so, the subproblem must be branched on this variable. New subproblems are generated by fixing integer values to variable x_i from its possible range. The new subproblems are generated as children nodes of variable x_i.

This algorithm repeats until no new nodes can be generated, because no fractional decision variables can be used to branch.

Fig. 2 shows a MIP problem example. The MIP model of the problem is defined in Fig. 2(a) and a graphical representation of the generated tree is shown in Fig. 2(b). The Branch & Bound starts applying Simplex to this model, obtaining the upper bound value of 92 for the objective function. The algorithm creates the root node (node 1 in Fig. 2(b)) of the tree with this value as the estimated value for the objective function ($E = 92$). Current value of the objective function for the integer solution is 0 due to no integer solution is known yet ($V = 0$). Constraint is not updated because decision variables are equal to 0 ($C = 0$). Simplex algorithm has solved the relaxed version of this problem obtaining $x_1 = 1, x_2 = 0.25$ and $x_3 = 1$. Variable x_2 is fractional at optimality, therefore, we must set an integer value for this variable. The algorithm sets the variable x_2 to 0 and then use Simplex with this information. Thus, the node 2 is created. The new solution is integer at optimality ($x_1 = 1, x_2 = 0, x_3 = 1$) obtaining an objective function value of 80. The new integer solution satisfies all the constraints and is better than the previous one (because no previous integer solution had been obtained), therefore, this solution is stored.

The algorithm also sets variable x_2 to value 1 and solve the problem using Simplex with this new constraint, and the node 3 is generated. The results of this solution are $x_1 = 0, x_2 = 1$ and $x_3 = 0.6667$ for decision variables and the estimated objective function value is $E = 71.33$. The best integer solution obtained has an objective function value of 80, hence, the new node solution (node 3) can be pruned, because no children of this subproblem could improve the objective function. The algorithm stops because

0									
	0			1					
		0							
			0						
				0					
					0				
						0			
							0		
								0	
									0

(a) X matrix search space reduction

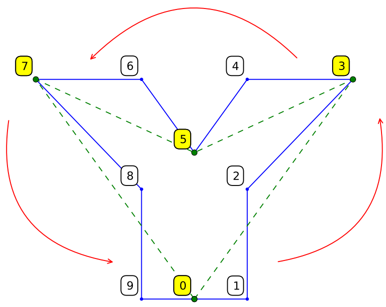
0	?								
	0			1					
	?	0							
	?		0						
?	?	?	?	0	?	?	?	?	?
	?				0				
	?					0			
	?						0		
	?							0	
	?								0

(b) Possible values in X matrix

Fig. 4. Fig. 4(a) represents the status of matrix X after a segment from point 1 to point 4 is selected. The constraint defined in Eq. (11) forces the model to take some of the values highlighted in Fig. 4(b).

0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

(a) Matrix X representation

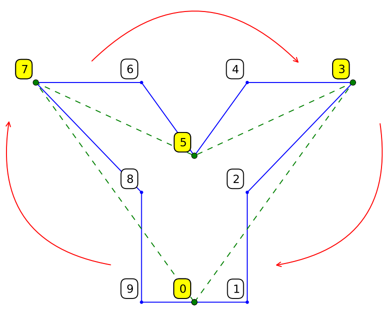


(b) Polygonal approximation

Fig. 5. Solution A (counterclockwise direction).

0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

(a) Matrix X representation



(b) Polygonal approximation

Fig. 6. Solution B (clockwise direction).

no new nodes could be generated. The best integer solution obtained in node 2, with values $x_1 = 1, x_2 = 0$ and $x_3 = 1$; is the optimal solution for this problem. A flowchart of this basic Branch & Bound algorithm is shown in Fig. 3.

Mixed Integer Programming technique has successfully been used in several fields: allocation of distributed generators in radial distribution systems [19], data envelop analysis [21], modern semiconductor manufacturing systems [1], etc.

2.1. Proposed MIP model

As explained above, the key point for solving a problem using MIP techniques is to create a suitable model that defines the problem. In this work, a novel MIP model for solving the problem of obtaining the optimal polygonal approximation for M points is proposed. Our model defines a matrix of binary decision variables X of size $N \times N$ (where N is the number of points of the contour). We

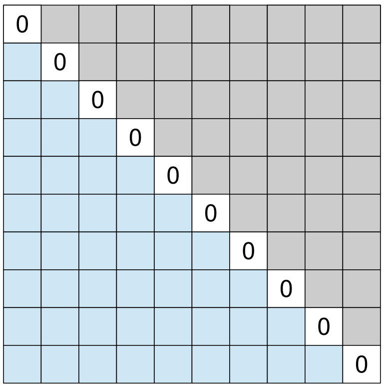


Fig. 7. X matrix symmetric spaces.

express the model using a matrix of decision variables in order to simplify the explanations. A binary decision variable x_{ij} is set to 1 when the segment that starts at contour's point i and ends at contour's point j is selected for the solution. Thus, we can express all possible solutions with matrix X. On the other hand, we define a matrix of coefficients Δ of size $N \times N$, where an element δ_{ij} is set to the distortion (we use the ISE value) associated to the segment defined by the element x_{ij}, which is calculated as appear in Eq. (1). The objective function for the proposed model is defined as:

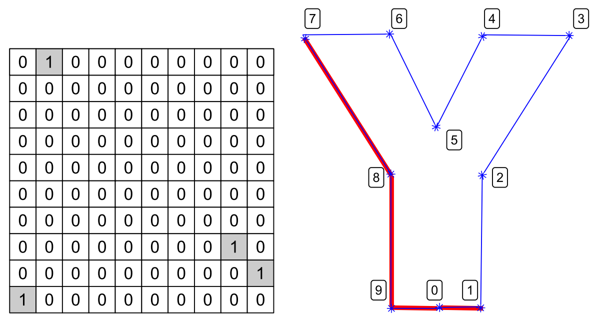
$$z = \min \sum_{i=0}^{N-1} \sum_{\substack{j=0 \\ j \neq i}}^{N-1} \delta_{ij} x_{ij} \tag{9}$$

This objective function calculates the distortion value (ISE) produced by a given solution represented by the matrix X. Note that the decision variables where $i = j$ are not taken into account because segments from point i to i does not make sense.

In order to obtain a feasible configuration of the binary matrix X, and therefore, a feasible optimal solution to the problem; a set of lineal constraints has to be supplied. The optimal solution provided must have a fixed number M of points, therefore, a constraint must be added:

$$\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x_{ij} = M \tag{10}$$

Moreover, if segment consisted of points i and j is selected for the solution (item (i,j) of matrix X is equal to 1), then some segment which ends with point i must be selected (some value of column i must be 1); and some segment which starts with point j must also be selected for the current solution (some value of row j must be 1). These restrictions can be modeled as follows:



(a) Matrix X representation (b) Polygonal approximation

Fig. 8. Not valid optimal solution for model without constraint defined in Eq. (11).

$$\sum_{r=0}^{N-1} x_{ir} = \sum_{c=0}^{N-1} x_{ci}, \forall i \in [0, N-1] \tag{11}$$

An example of this appear in Fig. 4(a), where the segment consisted of points 1 and 4 is selected for the solution. Taking into account the set of constraints defined in Eq. (11) some of the values in column 1 must be set to 1 (because summation of row 1 is equal to 1), and some of the values row 4 must be set to 1 (because summation of column 4 is equal to 1). This is represented in Fig. 4(b).

Finally, the proposed model may represent optimal solutions in two equivalent ways, as appear in Figs. 5 and 6. These two matrix configurations (Figs. 5(a) and 6(a)) are different, but define the same polygonal approximation in either counterclockwise (Fig. 5(b)) or clockwise direction (Fig. 6(b)).

Formally, a polygonal approximation of a curve S defined in counterclockwise direction is represented with an ordered number of points M, such as, their indexes $I = \{i_0, i_1, \ldots, i_{M-1}\}$ satisfy $i_0 < i_1 < \ldots < i_{M-1}$. Due to we represent a closed polygonal approximation, all the segments of the approximation are defined $(\overline{s_{i_0}s_{i_1}}, \overline{s_{i_1}s_{i_2}}, \ldots, \overline{s_{i_{M-2}}s_{iM-1}}, \overline{s_{i_{M-1}}s_{i_0}})$. Therefore, $M-1$ segments $(\overline{s_{i_a}s_{i_b}})$ satisfy $i_a < i_b$ and one segment $(\overline{s_{i_{M-1}}s_{i_0}})$ satisfies $i_{M-1} > i_0$. These two groups of segments are represented in two symmetric spaces in matrix X, which are graphically represented in Fig. 7. Segments $\overline{s_{i_a}s_{i_b}}$ that satisfy $i_a < i_b$ are represented in the upper (gray) space in matrix X (Fig. 7); the segment $\overline{s_{i_{M-1}}s_{i_0}}$ that satisfies $i_{M-1} > i_0$ is represented in the lower (blue) space in matrix X.

This previous information is used in order to avoid symmetric solutions and, therefore, boost the execution of the method. As explained above, a polygonal approximation defined in counterclockwise direction uses $M-1$ segments in the upper space in matrix X and one in the lower space. This information is used to create a constraint for symmetry breaking[2], that can be written as follows:

$$\sum_{i=0}^{N-1} \sum_{j=0}^{i} x_{ij} = 1 \tag{12}$$

2.2. Analysis of the model

The proposed model has been presented in Section 2.1. The proposed MIP model is defined using a set of constraints (Eqs. (10)–(12)). This set of constraints are used to define the problem and to obtain a feasible optimal solution for the polygonal approximation problem. But, are all of these constraints mandatory to define the model?

Let's suppose we have deleted the constraint presented in Eq. (10). This constraint forces the solver to select a number M of segments for the optimal solution. If this constraint is not used in the model, the trivial solution where all decision variables are equal to 0 will always satisfy all constraint. This trivial solution has a value of 0 for the objective function. Therefore, this constraint is necessary to define a suitable model for this problem.

The constraint defined in Eq. (11) forces the solver to select a closed polygonal approximation. If this constraint is deleted, the

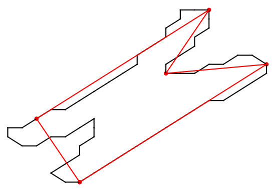


Fig. 9. The figure shows the chromosome contour. The optimal polygonal approximation for $M = 5$ is highlighted.

Table 1
The table summarizes the steps of the algorithm Branch & Bound using the proposed model. The model obtains the polygonal approximation for contour chromosome for $M = 5$.

Node	Variable	Value	Function value	Integrality constraint	Action
1	$x_{15,22}$	1	79.03	Yes	Save this solution as the best integer solution found
2	$x_{15,22}$	0	79.91	No	Prune due to worse objective function
3	$x_{15,23}$	1	80.05	No	Prune due to worse objective function
4	$x_{15,23}$	0	79.03	Yes	Prune due to equal objective function
5	$x_{22,29}$	1	79.03	Yes	Prune due to equal objective function
6	$x_{22,29}$	0	79.91	No	Prune due to worse objective function
7	$x_{23,52}$	1	130.35	No	Prune due to worse objective function
8	$x_{23,52}$	0	79.03	Yes	Prune due to equal objective function
9	$x_{29,47}$	1	89.96	Yes	Prune due to worse objective function
10	$x_{29,47}$	0	79.03	Yes	Prune due to equal objective function
11	$x_{47,54}$	1	94.42	Yes	Prune due to worse objective function
12	$x_{47,54}$	0	79.03	Yes	Prune due to equal objective function
13	$x_{52,0}$	1	79.95	No	Prune due to worse objective function
14	$x_{52,0}$	0	79.03	Yes	Prune due to equal objective function
15	$x_{54,0}$	1	93.74	Yes	Prune due to worse objective function
16	$x_{54,0}$	0	79.03	Yes	Prune due to equal objective function

solver may select segments that do not form a valid polygonal approximation. For example, let's use this model (without Eq. (11)) to obtain the optimal polygonal approximation with $M = 4$ for the contour that appears in Fig. 5(b). The MIP solver obtains a optimal solution where $x_{0,1} = x_{7,8} = x_{8,9} = x_{9,0} = 1$ and objective function is equal to 0. This solution satisfies constraints defined in Eqs. (10) and (12). Fig. 8 shows the optimal solution obtained for this model. The solution highlighted in Fig. 8(b) does not form a polygon, therefore, the constraint defined in Eq. (11) forces the solver to take closed polygonal approximations, instead of isolated line segments.

Finally, the constraint defined in Eq. (12) is used to break symmetries in the model. A MIP problem is called symmetric if the variables can be permuted without changing the structure of the problem[2]. This type of models can represent feasible optimal solutions in equivalent ways. Section 2.1 demonstrates that the proposed model can represent optimal feasible solutions in counterclockwise and clockwise direction. These symmetries should be avoided, due to the additional computational burden required by the Branch & Bound algorithm to explore the equivalent solutions[2]. This problem is solved by adding the constraint defined in Eq. (12).

This model defines the problem and the constraints manage to obtain a feasible optimal solution. This is achieved by using the Branch & Bound algorithm, as explained in Section 2. In order to understand the whole process, a small example is shown.

Let's suppose that we want to obtain the polygonal approximation for contour chromosome, Fig. 9, with $M = 5$. The process starts applying the Simplex algorithm to the proposed model with no integer restrictions. The algorithm obtains a value for the objective function equal to 78.92. The decision variable $x_{0,15}$ has a integer value of 1, variables $x_{15,22}, x_{15,23}, x_{22,29}, x_{23,52}, x_{29,47}, x_{47,54}, x_{52,0}, x_{54,0}$ are equal to 0.5, and the rest of the variables are equal to 0.

The Branch & Bound algorithm enumerates all possible values, 1 or 0, in the present model, for decision variables which are not integer. The value of 1 is fixed for the variable $x_{15,22}$ and the Simplex algorithm is executed with this new constraint. The node 1 is created in the search tree. The objective function value obtained is equal to 79.03 and all the constraints are satisfied, therefore, this solution is stored as the best solution found. The algorithm also fixes variable $x_{15,22}$ to value 0 and creates node 2 in the search tree. The Simplex algorithm is executed using this constraint. The integrality constraint is not satisfied, that is, some decision variable is fractional, but the objective function is equal to 79.91 and this node is pruned due to no children of this subproblem could improve the best feasible solution found.

The Branch & Bound selects next decision variable, $x_{15,23}$, and fixes value to 1. The node 3 is created in the search tree. The Simplex algorithm is executed using this new constraint. The objective function value obtained is 80.05, therefore, the node is pruned because no children of this node could improve the best solution found. The variable is also fixed to value 0 and Simplex

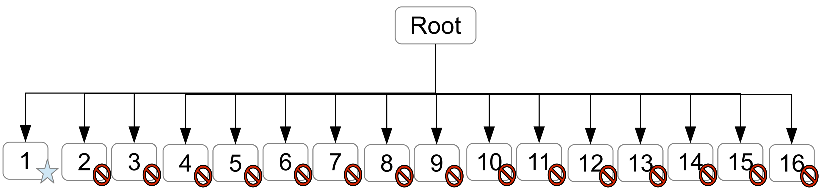


Fig. 10. Search tree generated by the Branch & Bound algorithm tree. The optimal solution has been found in the node 1. The rest of the solutions found are pruned due to the objective function value is not improved.

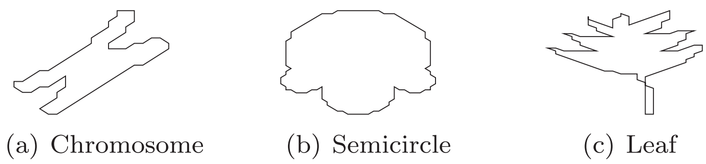


Fig. 11. Synthetic contours.

is executed taking into account this new constraint. The node 4 is created in the search tree. The integrality constraint is satisfied, that is, all decision variables are integer. The objective function value is equal to 79.03, therefore, this solution is not stored because does not improve the value (the value is equal to the best solution found).

This search process continues until no new nodes are generated, as explained in Section 2. A summary of this process is present in Table 1. The search process has found the optimal solution in node 1. The rest of nodes have been pruned due to the objective function values do not improve the optimal value found in node 1. Therefore, no new levels of the search tree have been added. The generated search tree is shown in Fig. 10. The polygonal approximation and the original digital curve appear in Fig. 9.

3. Experiments and results

This section describes the experiments carried out in order to demonstrate the advantages of using this procedure to calculate the optimal polygonal approximation of a digital planar curve. Three experiments have been carried out comparing several methods and using different contours. In the first experiment we check the optimality of the solution for several methods. In the second experiment we compare computation times for methods that assure optimal solutions. The third experiment shows the speedup obtained using the present proposal.

For the comparisons we have considered the unique algorithms that assure optimal solutions [17,20] among the literature and some algorithms which obtain polygonal approximations close to the optimal solution [11,15,5,16,3].

The experiments have been carried out in a generic computer with processor Intel(R) Core(TM) i7-3930K CPU @ 3.20 GHz, 16 GB of RAM memory. The present proposal has been developed using C++ as programming language and using Gurobi [10] (version 5.6.3) as the LP/MIP solver library. The default configuration for the MIP solver library was used. Gomory's mixed integer cuts [8] has also been enabled in order to boost the process. All the compared methods were implemented and run according to the specifications present in their respective papers. The method proposed by Salotti [20] was supplied by the author.

3.1. Comparisons between optimal and suboptimal algorithms

The first experiment has been carried out to determine the optimality of the solutions provided by some methods. As discussed in Section 1 the method proposed by Perez and Vidal [17] is usually used to compute the optimal polygonal approximation. However, an initial point must be set as a parameter. We have used the common standard defined in the literature of using the first point of the curve as the initial point for this purpose. We refer this version of the algorithm as Perez-Vidal$_0$. As is mentioned in Section 1 this may produce suboptimal solutions, therefore, we have tried all points as initial point. We refer to this way of using the algorithm as Perez-Vidal$_N$. We evaluate the method proposed by Salotti [20], which also must try all initial points to obtain the optimal solution (Salotti$_N$). We also evaluate the methods proposed by Horng and Li [11], Masood [15], Carmona-Poyato et al. [5], Parvez and Mahmoud [16], Backes and Bruno [3] and our proposal.

The synthetic contours chromosome, semicircle and leaf have been used in this experiment. These contours are broadly used in the literature as a standard benchmark to compare polygonal approximation algorithms[22,4,3,11,15]. For each synthetic contour, several polygonal approximations with different size *M*, have been obtained. The synthetic contours are shown in Fig. 11.

Table 2 shows the results of the first experiment. These values demonstrates that the optimal solution is always obtained for methods Perez-Vidal$_N$, Salotti$_N$ and our proposal. The results show that method by Horng and Li [11], Masood [15], Carmona-Poyato et al. [5], Parvez and Mahmoud [16] and the method by Backes and Bruno [3] obtains suboptimal results several times, hence,

Table 2
This table shows the results for methods Perez-Vidal (using the first point of the curve as initial point Perez-Vidal$_0$, and trying all points as initial point Perez-Vidal$_N$), Salotti [20] (using the first point of the curve as initial node Salotti$_0$, and trying all points Salotti$_N$), Horng and Li [11], Masood [15], Carmona-Poyato et al. [5], Parvez and Mahmoud [16], Backes and Bruno [3] and our proposal. These results show that the unique alternatives that obtain the optimal solution in every situation are Perez-Vidal$_N$, Salotti$_N$ and our proposal.

Contour	Method	M	ISE	Optimal
Chromosome	Parvez-Mahmoud	10	14.34	No
	Proposed	10	8.0680	Yes
	Backes-Bruno	11	7.8	No
	Proposed	11	7.06	Yes
	Perez-Vidal$_0$	12	6.06	No
	Perez-Vidal$_N$	12	5.82	Yes
	Salotti$_0$	12	6.06	No
	Salotti$_N$	12	5.82	Yes
	Horng	12	5.82	Yes
	Masood	12	5.82	Yes
	Backes-Bruno	12	5.82	Yes
	Proposed	12	5.82	Yes
	Perez-Vidal$_0$	15	3.97	No
	Perez-Vidal$_N$	15	3.79	Yes
	Salotti$_0$	15	3.97	No
	Salotti$_N$	15	3.79	Yes
	Horng	15	3.81	No
	Masood	15	3.88	No
	Carmona-Poyato	15	4.27	No
	Proposed	15	3.79	Yes
Semicircle	Backes-Bruno	12	28.9	No
	Proposed	12	26.0045	Yes
	Perez-Vidal$_0$	15	14.40	Yes
	Perez-Vidal$_N$	15	14.40	Yes
	Salotti$_0$	15	14.40	Yes
	Salotti$_N$	15	14.40	Yes
	Horng	15	14.40	Yes
	Masood	15	14.40	Yes
	Backes-Bruno	15	14.40	Yes
	Proposed	15	14.40	Yes
	Parvez-Mahmoud	17	19.02	No
	Proposed	17	12.2179	Yes
	Perez-Vidal$_0$	25	4.75	No
	Perez-Vidal$_N$	25	4.62	Yes
	Salotti$_0$	25	4.75	No
	Salotti$_N$	25	4.62	Yes
	Horng	25	4.75	No
	Masood	25	4.75	No
	Proposed	25	4.62	Yes
	Carmona-Poyato	26	4.91	No
	Proposed	26	4.0543	Yes
Leaf	Backes-Bruno	12	70.5	No
	Proposed	12	50.1983	Yes
	Perez-Vidal$_0$	13	55.88	No
	Perez-Vidal$_N$	13	38.32	Yes
	Salotti$_0$	13	55.88	No
	Salotti$_N$	13	38.32	Yes
	Horng	13	42.05	No
	Masood	13	64.85	No
	Proposed	13	38.32	Yes
	Parvez-Mahmoud	21	13.82	No
	Proposed	21	8.94993	Yes
	Carmona-Poyato	23	10.68	No
	Proposed	23	7.46609	Yes
	Perez-Vidal$_0$	32	4.45	No
	Perez-Vidal$_N$	32	3.45	Yes
	Salotti$_0$	32	4.45	No
	Salotti$_N$	32	3.45	Yes
	Horng	32	4.45	No
	Masood	32	4.45	No
	Proposed	32	3.45	Yes

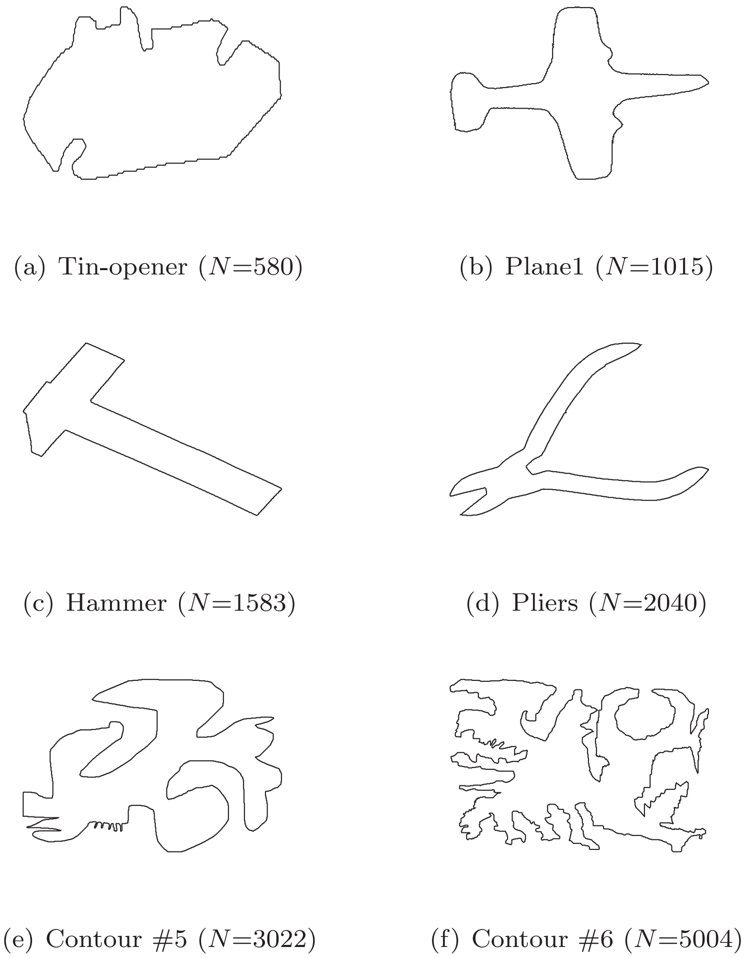


Fig. 12. Contours used in experiment 3.

Table 3
The computation time obtained in seconds for method Perez-Vidal$_N$, Salotti$_N$ and our proposal. The results show that our proposal obtains the optimal solution taking lower computation time than the other methods.

Contour	Method	*M*				
		10	20	30	40	50
Tinopener	Perez-Vidal$_N$	13.92	23.2	25.52	27.84	30.16
	Salotti$_N$	6.7	7.61	8.5	7.12	11.22
	Proposed	5.03	4.54	4.33	4.14	6.96
Plane1	Perez-Vidal$_N$	85.26	121.8	178.64	219.25	259.87
	Salotti$_N$	37.23	30.45	39.5	42.85	43.31
	Proposed	19.51	15.00	13.99	14.77	13.53
Hammer	Perez-Vidal$_N$	291.27	525.56	740.85	918.14	1108.1
	Salotti$_N$	142.3	131.74	179.23	183.54	215.23
	Proposed	47.01	40.33	38.10	43.25	59.38
Pliers	Perez-Vidal$_N$	669.12	1175.04	1648.32	2088.96	2586.72
	Salotti$_N$	331.02	389.61	395.25	315.79	323.56
	Proposed	94.89	77.72	72.32	67.55	64.21
Contour #5	Perez-Vidal$_N$	2546.41	4437.72	6240.98	8030.16	9791.16
	Salotti$_N$	1061.01	964.72	1101.35	1159.91	1073.59
	Proposed	369.49	488.14	254.92	287.82	230.08
Contour #6	Perez-Vidal$_N$	9307.44	16012.80	22818.24	30024	36379.08
	Salotti$_N$	5318.53	4927.01	4563.65	4740.63	4921.87
	Proposed	1682.45	909.45	915.03	790.70	867.86

these methods do not assure the optimal solution. Note that using the first point on the curve as the initial point in the methods by Perez-Vidal and Salotti (Perez-Vidal$_0$ and Salotti$_0$) results in sub-optimal solutions, as appear in Table 2.

3.2. Comparisons between optimal algorithms

The proposed method based on MIP techniques obtains the optimal polygonal approximation for M points, therefore, we

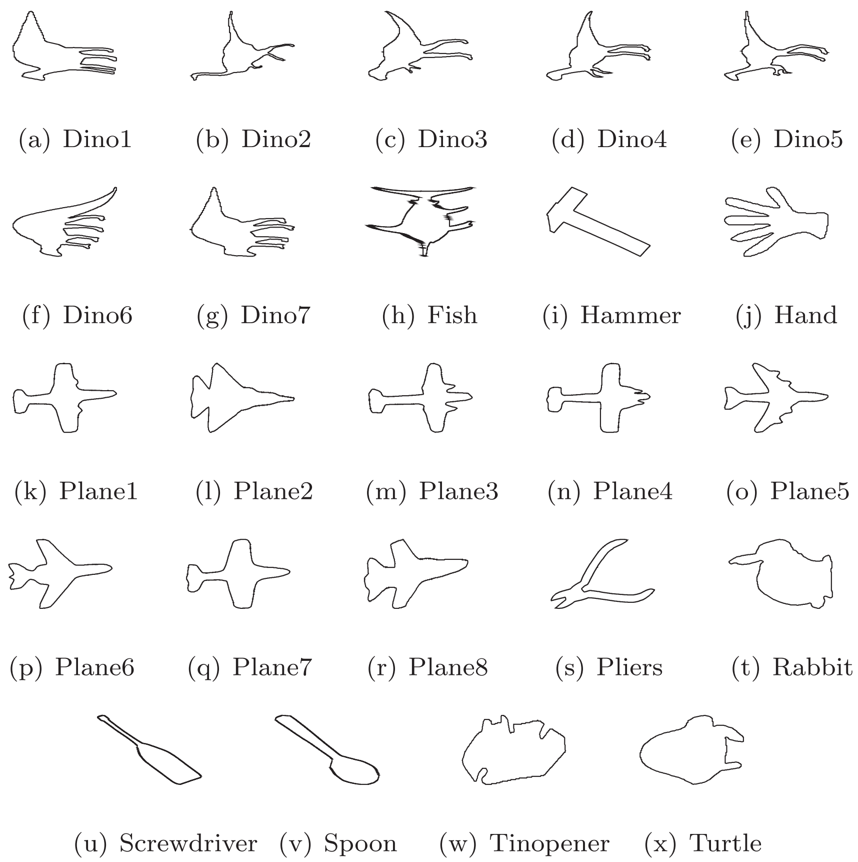


Fig. 13. Real contours dataset.

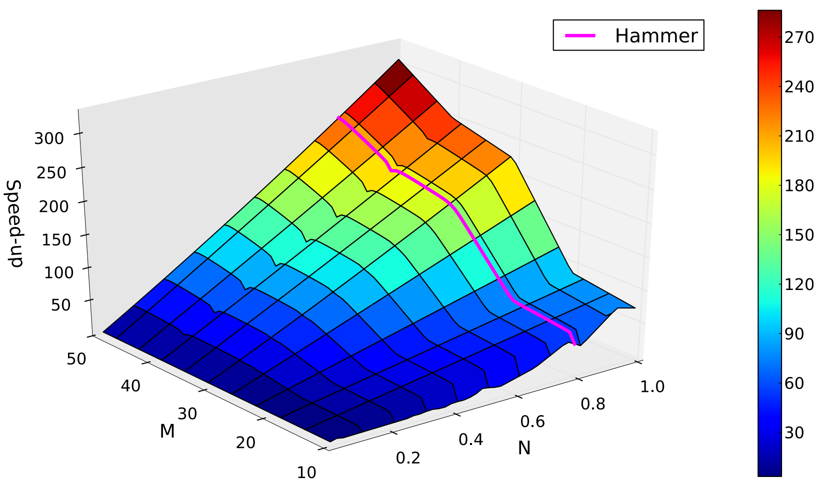


Fig. 14. The figure presents how the speedup (Speedup $= \frac{\text{Time}_{\text{Perez-Vidal}}}{\text{Time}_{\text{MIP}}}$) raises with increasing number of points N of the contour and number of points M of the polygonal approximation. The value N is normalized between the minimum and maximum number of points of the contours used in the experiment. The size of the polygonal approximation M is represented as a percentage of the break points of the contour. The speedup obtained for the contour known as hammer is highlighted.

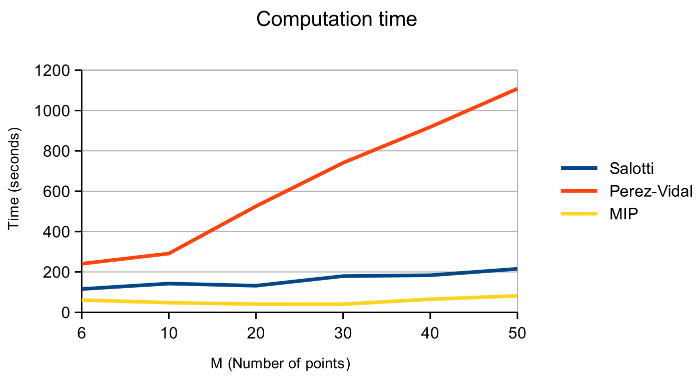
compare this method with the alternative proposed by Perez and Vidal [17] and the method proposed by Salotti [20] because are the unique alternatives which can obtain the optimal solution for closed curves, as shown in Section 3.1.

A set of real contours, shown in Fig. 12, have been used in this experiment (contour #5 and contour #6 are provided by Salotti [20]). These contours have been used as the input for the method by Perez-Vidal, the alternative proposed by Salotti and the present MIP model. We have obtained polygonal approximations with sizes $M = 10$, 20, 30, 40 and 50.

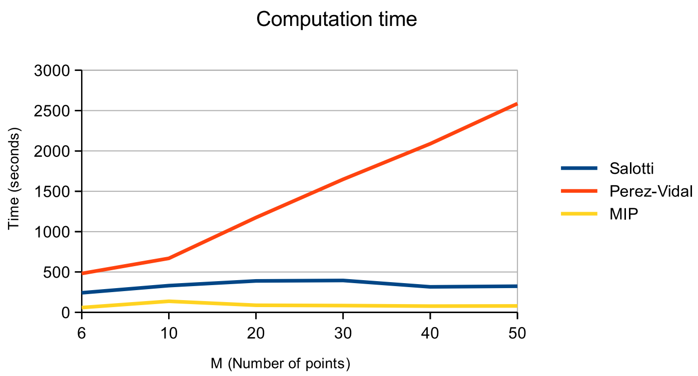
The results in Table 3 show that the computation time obtained by our proposal is lower than the computation time obtained by Perez and Vidal [17] and Salotti [20]. The difference between the computation times for the different methods is remarkable.

3.3. Comparing the speedup obtained

The differences shown in Table 3 between the method by Perez and Vidal [17] and our proposal increases with increasing number of points M of the polygonal approximation. We have performed



(a) Hammer



(b) Pliers

Fig. 15. These figures present the computation time evolution for obtaining optimal polygonal approximations of two contours. The computation time increments with increasing number of points M of the polygonal approximation for the alternative proposed by Perez-Vidal. The figures show that the computation time for MIP solution does not change significantly for different number of points M of the polygonal approximation. The figures also show that the computation time by the proposed method is significantly lower than the other alternatives presented in the figures.

another experiment in order to verify if this trend continues. Fig. 13 shows the dataset used for this purpose, which includes 24 real contours. We have used five sizes for the polygonal approximations (M = 10%, 20%, 30%, 40% and 50% of the breakpoints of the contour). For each contour and size M of polygonal approximation the computation times for the method based on Dynamic Programming and our proposal have been obtained. We have computed the speedup ($\text{Speedup} = \frac{\text{Time}_{\text{Perez-Vidal}}}{\text{Time}_{\text{MIP}}}$) using these computation times. Fig. 14 shows that the trend continues and the differences between the method based on Dynamic Programming and the proposed method keep increasing.

4. Discussion

In this section some relevant aspects of the proposed method are discussed.

One of the drawbacks of the alternative proposed by Perez-Vidal and the method proposed by Salotti is that the initial point of the polygonal approximation is required, as is mentioned in Section 1. The proposed method to obtain the optimal polygonal approximation does not require an initial point to be set as a parameter into the MIP model, that is, the method obtains the optimal solution by enumerating all feasible combinations and pruning those solutions which are worse that the best solution found, as explained in Section 2.

The main problem for obtaining the optimal polygonal approximation is that it is a computationally expensive process. This issue, appears in Table 3, where computation times grow with increasing number N of points of the original contour for the Dynamic Programming alternative, the method by Salotti [20] and our proposal. Although the computation time of all optimal methods used in the experiments depends on the number N of segments of the original contour, the results shown in Table 3 demonstrates that the computation time obtained by our proposal is significantly lower than the time obtained by both the alternative by Perez-Vidal and the method proposed by Salotti [20].

The method proposed by Perez-Vidal presents another problem. The computation time increases with increasing number of points M in the polygonal approximation. This problem is not present in the MIP solution as Fig. 15(a) and (b) show. The figures show that the computation time for the alternative by Perez-Vidal increases as the number of points M of the polygonal approximation also increases. On the other hand, the computation time for the proposed MIP alternative does not change significantly with increasing number of points M of the polygonal approximation. These differences have been shown to increase with increasing number of points N of the contour and also with increasing number of points M of the polygonal approximation. This trend is shown in Fig. 14, where the speedup over the method based on Dynamic Programming keep increasing. The proposed method has achieved computation times up to 300 faster than the method based on Dynamic Programming. The differences between the proposed method and the method proposed by Salotti [20] are also remarkable as appear in Table 3. Our proposal has obtained computational times up to 6 times faster than the method introduced by Salotti.

5. Conclusions

A novel method to compute optimal polygonal approximations of closed curves based on Mixed Integer Programming has been presented in the present paper. This method presents several advantages over the state-of-the-art alternatives, namely:

- The proposed method does not need the initial point of the polygonal approximation to compute the optimal polygonal approximation.
- The computation time does not depends on the number of points *M* of the polygonal approximation.
- The computation time needed to obtain the optimal polygonal approximation is significantly lower than the time required by the state-of-the-art alternatives.

Since the computation time of our method is modest it may be used for applications that need the optimal polygonal approximation such as the computation of the merit.

Acknowledgments

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

We would like to thank Jean-Marc Salotti for providing the software and contours used in the experiments of this paper.

References

[1] Adrián M. Aguirre, Carlos A. Méndez, Gloria Gutierrez, Cesar De Prada, An improvement-based MILP optimization approach to complex AWS scheduling, Comput. Chem. Eng. 47 (2012) 217–226, http://dx.doi.org/10.1016/j.compchemeng.2012.06.036. ISSN: 0098-1354.

[2] J. Alemany, F. Magnago, D. Moitre, H. Pinto, Symmetry issues in mixed integer programming based unit commitment, Int. J. Electr. Power Energy Syst. 54 (2014) 86–90, http://dx.doi.org/10.1016/j.ijepes.2013.06.034. ISSN: 0142-0615.

[3] André Ricardo Backes, Odemir Martinez Bruno, Polygonal approximation of digital planar curves through vertex betweenness, Inform. Sci. 222 (2013) 795–804, http://dx.doi.org/10.1016/j.ins.2012.07.062. ISSN: 0020-0255.

[4] A. Carmona-Poyato, N.L. Fernández-García, R. Medina-Carnicer, F.J. Madrid-Cuevas, Dominant point detection: a new proposal, Image Vision Comput. 23 (13) (2005) 1226–1236, http://dx.doi.org/10.1016/j.imavis.2005.07.025. ISSN: 0262-8856.

[5] A. Carmona-Poyato, F.J. Madrid-Cuevas, R. Medina-Carnicer, R. Muñoz Salinas, Polygonal approximation of digital planar curves through break point suppression, Pattern Recogn. 43 (1) (2010) 14–25, http://dx.doi.org/10.1016/j.patcog.2009.06.010. ISSN: 0031-3203.

[6] George Bernard Dantzig, Linear Programming and Extensions, Princeton University Press, 1965. ISBN: 0691059136.

[7] David H. Douglas, Thomas K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Cartographica: Int. J. Geographic Inform. Geovisual. 10 (2) (1973) 112–122, http://dx.doi.org/10.3138/FM57-6770-U75U-7727.

[8] Ralph E. Gomory, Outline of an algorithm for integer solutions to linear programs, Bull. Am. Math. Soc. 64 (5) (1958) 275–278.

[9] K. Grauman, T. Darrell, Fast contour matching using approximate earth mover's distance, in: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004 (CVPR 2004), vol. 1, 2004, pp. I-220–I-227. doi:http://dx.doi.org/10.1109/CVPR.2004.1315035.

[10] Inc. Gurobi Optimization, Gurobi optimizer reference manual, 2014. <http://www.gurobi.com>.

[11] Ji-Hwei Horng, Johnny T. Li, An automatic and efficient dynamic programming algorithm for polygonal approximation of digital curves, Pattern Recogn. Lett. 23 (1–3) (2002) 171–182, http://dx.doi.org/10.1016/S0167-8655(01)00098-8. ISSN: 0167-8655.

[12] Hiroshi Imai, Masao Iri, Polygonal approximations of a curve formulations and algorithms, in: Godfried T. Toussaint (Ed.), Machine Intelligence and Pattern Recognition, Computational Morphology A Computational Geometric Approach to the Analysis of Form, vol. 6, North-Holland, 1988, pp. 71–86. ISBN: 0923-0459.

[13] A.H. Land, A.G. Doig, An automatic method for solving discrete programming problems, Econometrica 28 (3) (1960) 497–520.

[14] David G. Lowe, Three-dimensional object recognition from single two-dimensional images, Artif. Intell. (1987) 355–395.

[15] Asif Masood, Optimized polygonal approximation by dominant point deletion, Pattern Recogn. 41 (1) (2008) 227–239, http://dx.doi.org/10.1016/j.patcog.2007.05.021. ISSN: 0031-3203.

[16] Mohammad Tanvir Parvez, Sabri A. Mahmoud, Polygonal approximation of digital planar curves through adaptive optimizations, 31(13), 1997–2005. ISSN: 0167-8655. doi:http://dx.doi.org/10.1016/j.patrec.2010.06.007.

[17] Juan-Carlos Perez, Enrique Vidal, Optimum polygonal approximation of digitized curves, Pattern Recogn. Lett. 15 (8) (1994) 743–750, http://dx.doi.org/10.1016/0167-8655(94)90002-7. ISSN: 0167-8655.

[18] P.L. Rosin, Techniques for assessing polygonal approximations of curves, IEEE Trans. Pattern Anal. Machine Intell. 19 (6) (1997) 659–666, http://dx.doi.org/10.1109/34.601253. ISSN: 0162-8828.

[19] Augusto C. Rueda-Medina, John F. Franco, Marcos J. Rider, Antonio Padilha-Feltrin, Rubén Romero, A mixed-integer linear programming approach for optimal type, size and allocation of distributed generation in radial distribution systems, Electric Power Syst. Res. 97 (2013) 133–143, http://dx.doi.org/10.1016/j.epsr.2012.12.009. ISSN: 0378-7796.

[20] Marc Salotti, An efficient algorithm for the optimal polygonal approximation of digitized curves, Pattern Recogn. Lett. 22 (2) (2001) 215–221, http://dx.doi.org/10.1016/S0167-8655(00)00088-X. ISSN: 0167-8655.

[21] Ying-Ming Wang, Peng Jiang, Alternative mixed integer linear programming models for identifying the most efficient decision making unit in data envelopment analysis, Comput. Indust. Eng. 62 (2) (2012) 546–553, http://dx.doi.org/10.1016/j.cie.2011.11.003. ISSN: 0360-8352.

[22] Wen-Yen Wu, Dominant point detection using adaptive bending value, Image Vision Comput. 21 (6) (2003) 517–525, http://dx.doi.org/10.1016/S0262-8856(03)00031-3. ISSN: 0262-8856.

Chapter 4

Third contribution: New method for obtaining optimal polygonal approximationsto solve the min-ε problem

Neural Comput & Applic
DOI 10.1007/s00521-016-2198-7

IBPRIA 2015

New method for obtaining optimal polygonal approximations to solve the min-ε problem

A. Carmona-Poyato[1] · E. J. Aguilera-Aguilera[1] · F. J. Madrid-Cuevas[1] · M. J. Marín-Jiménez[1] · N. L. Fernández-García[1]

Received: 30 September 2015 / Accepted: 12 January 2016

Abstract A new method for obtaining optimal polygonal approximations in closed curves is proposed. The new method uses the suboptimal method proposed by Pikaz and an improved version of the optimal method proposed by Salotti. Firstly, the Pikaz's method obtains a suboptimal polygonal approximation and then the improved Salotti's method is used for obtaining many local optimal polygonal approximations with a prefixed starting point. The error value obtained in each polygonal approximation is used as value of pruning to obtain the next polygonal approximation. In order to select the starting point used by the Salotti's method, five procedures have been tested. Tests have shown that by obtaining a small number of polygonal approximations, global optimal polygonal approximation is calculated. The results show that the computation time is significantly reduced, compared with existing methods.

Keywords Digital planar curve · Global optimal polygonal approximation · Local optimal polygonal approximation · Pikaz's method · Perez's method · Salotti's method

1 Introduction

Polygonal approximations of digital planar curves are an important problem in image processing, pattern recognition and computer graphics. They are used in applications like image analysis [1], shape analysis [2], geographical information systems [3] and digital cartography [4]. The main goal is to provide a compact representation of the original curve with reduced memory requirements, preserving the important shape information.

The problem can be defined as follows: given a digital planar curve C with N points, approximate it by an other digital planar curve C_a with a prefixed number of points M so that the obtained error in this approximation is minimized. This problem is known as min-ε problem or minimum-distortion problem. To solve this problem in an optimal way, many methods have been proposed: (i) using graph theory [5–7], (ii) using dynamic programming [8, 9] and (iii) using A*-search algorithm [10].

These methods solve this problem in $O(N^2) - O(MN^2)$ time in open curves or closed curves when a starting point is prefixed. The main drawback of the cited methods is due to the 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 of complexity and becomes $O(N^3) - O(MN^3)$.

Kolesnikov and Franti [11] proposed a new method based on reduced search approach that provides a solution very close to the optimal one; however, the global optimal polygonal approximation is not guaranteed.

There are many suboptimal methods that solve this problem; for example, the method proposed by Pikaz and Dinstein [12] is $O(n \log n)$, though they are faster than optimal methods but they are not optimal.

In this work, a method based on the Pikaz's method [12] and the Salotti's method [10] is proposed to solve the min-ε problem. Its computational complexity is close to $O(N^2)$

✉ A. Carmona-Poyato
ma1capoa@uco.es

[1] Department of Computing and Numerical Analysis, Córdoba University, 14071 Córdoba, Spain

Published online: 04 February 2016

and between $O(N^2)$ and $O(MN^2)$. Therefore the computational complexity of the previously cited methods is reduced one level.

In Sect. 2 the main related methods are described. The proposed method is explained in Sect. 3. The experimental results are shown in Sect. 4, and finally, the main conclusions are summarized in Sect. 5.

2 Related methods

2.1 Pikaz's suboptimal method

Pikaz and Dinstein [12] proposed a suboptimal method based on a greedy iterative algorithm that eliminates the break point with the smallest error value. Break points are taken as the initial polygonal approximation. To calculate the error associated with each break point BP_j, two neighboring break points, BP_{j-1} and BP_{j+1}, are joined with a straight line. Maximum perpendicular (squared) distance of all boundary points between BP_{j-1} and BP_{j+1} from the straight line is called as *associated error value* of break point BP_j. In each iteration, only the break point with the smallest error value is deleted. If more than one *BP* with the smallest associated error value exists, any of them may be removed because sequence of removal (in case of more than one candidate) will not affect the results. When a break point is deleted, only the error associated with its two neighbors must be updated. For example, if BP_j is deleted, only the associated error value corresponding to BP_{j-1} and to BP_{j+1} is updated. This method is very fast $(O(n \log n))$ and can produce polygonal approximation with any preset number of final points.

2.2 Perez's method

Perez and Vidal [8] proposed a method, based on dynamic programming, to solve the min-ε problem, when the considered error is the integral squared error (*ISE*). The value of *ISE* for a polygonal approximation is defined as

$$\mathrm{ISE} = \sum_{i=1}^{N} e_i^2 \quad (1)$$

where:

- e_i is the orthogonal distance from P_i to the approximated line segment.
- P_i depicts any point of the digital curve.

Dynamic programming is an optimization method which makes a decision based on all possible previous states with a proper recurrence relation. Perez et al. used the next recursive function to solve the min-ε problem:

$$E(N,M) = \min_{M-1 \le i \le N-1} \{E(i, M-1) + e(P_i, P_N)\}$$

where:

- $E(N, M)$ depicts the minimum error of approximate the first N points by using M points
- $E(i, M-1)$ depicts the minimum error of approximate the first i points by using $M-1$ points.
- $e(P_i, P_N)$ depicts the error of approximate the curve segment between P_i and P_N by a single edge.

If the values of $e(P_i, P_j)$ are calculated in an incremental way, the computational complexity of this method is $O(MN^2)$ in open planar curves or closed curves when the starting point is prefixed. However, in closed curves, when all the points should be considered as starting point, the computational complexity is $O(MN^3)$.

2.3 Salotti's method

Salotti [10] proposed a method based on the search of the shortest path in a graph using A^* algorithm, to solve the min-ε problem. If the A^* algorithm is applied to solve this problem, it is slower than Perez's method due to the cost of the management of the graph and node sorting. In order to reduce the search, Salotti [10] proposed two improvements:

- 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 reason, Salotti proposed a simple solution to stop 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:

  $$E_{\mathrm{low}}^{P_i \to P_j} = \frac{1}{2} \mathrm{Min}\left(E_{reg1}^{P_i \to P_j}, E_{reg2}^{P_i \to P_j}\right)$$

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

Using these improvements, Salotti managed to reduce the time complexity of the A^* algorithm. In this case the computational complexity is close to $O(N^2)$. Since 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 $O(N^3)$.

2.4 Horng's method

Horng and Li [9] proposed a method for obtaining the global optimal polygonal approximation without testing all the points of the closed curve as starting point. They used three techniques to reduce the time complexity of the dynamic programming algorithm proposed by Perez and Vidal [8]

- Incremental error measure to estimate $e(P_i, P_j)$ in a constant time [8].
- Error measure is reused when the starting point changes. So, unnecessary repeated computations are avoided, because each error measure is computed only once, although the starting point changes [13].
- Initial point determination. He proposed to apply dynamic programming using a random starting point and then applied dynamic programming using as starting point the vertex farthest from the starting point of the first iteration and separated from its nearest vertex by more than a given threshold [9]

Horng's method has an overall computational complexity of $O(MN^2)$. However it does not guarantee the global optimal solution, because the starting point in the second iteration might not belong to the global optimal polygonal approximation.

2.5 Kolesnikov's method

Kolesnikov and Franti [11] proposed a solution, based on the optimal programming algorithm for open curves, very close to the optimal one. This solution uses a cyclically extended closed curve of double size and selects the optimal starting point by search in the extended search space for the curve. Kolesnikov's method solves min-ε and min-# problems in a processing time between 1.5 and 2 times of the processing time from the open curve.

To reduce the time-consuming search in the state space, they use a reduced search algorithm [14]. This algorithm starts from an initial solution generated using any fast algorithm that defines a reference path in the state space. A bounding corridor of width w is used along this path, and then, the minimum cost path within the corridor is obtained using Perez's method [8].

Although this method improves Horng'method, it does not guarantee the global optimal solution and only four contours were used to test it.

3 Proposed method

The following features of the proposed method can be highlighted:

- The proposed method relies on an improvement of the Salotti method [15] to obtain a local optimal polygonal approximation using a prefixed starting point in closed curves. Thus the processing time is reduced.
- An iterative procedure to obtain the best starting point is used. In each iteration of this procedure a new local optimal polygonal approximation, using a new starting point, is obtained, and this local polygonal approximation is better than the previous one.
- In order to select the new starting point in each iteration, some proposals have been tested and the best of them has been selected.

The method is detailed in the following subsections.

3.1 Improved Salotti's method

To improve the Salotti method, we propose to calculate the lower bound using the minimum error of the best line segment approximating a set S of consecutive points $(P_i, \ldots, P_j)$ instead of using the linear regressions y / x and x / y to estimate least-square errors. This method is known as *total least squares* or *orthogonal regression*. By using this method the time taken to calculate the lower bound is halved. This improvement was used in a previous work of the authors and reduces the computation time of the original Salotti's method about 16 % [16].

3.2 Global optimal polygonal approximation

A local optimal polygonal approximation is the optimal polygonal approximation for a prefixed starting point. In order to select the best starting point and for obtaining the optimal approximation, our proposal is based on the following statement:

If we obtained the global optimal polygonal approximation (the best of all local optimal polygonal approximations) of M points for a planar digital curve of N points, any local optimal polygonal approximation that uses any of these M points as prefixed starting point would be a global optimal polygonal approximation.

Demonstration: *If we obtain a local optimal polygonal approximation of M points for a planar digital curve of N points, using a prefixed starting point, any local optimal polygonal approximation of M points obtained using any of the M points of this polygonal approximation will be equal or better than the first local optimal polygonal approximation. Therefore if we use any of the M points of the global optimal polygonal approximation, as starting point to obtain a local optimal polygonal approximation, this approximation will be equal to the global optimal polygonal approximation (it can not be better).*

Considering these statements, two proposals have been tested and the best one has been selected. Our proposals can be summarized as follows:

3.3 Perez–Salotti proposal (*PS*)

1. Select a random starting point.
2. Obtain the local polygonal approximation for this starting point using the Perez's method. This method is used because in the first iteration no value of pruning is used and in this case the Perez method is faster. Obviously, this is not the global optimum.
3. Select the starting point for the second and next iterations. Horng and Li [9] propose only two iterations, and he used as starting point the vertex farthest from the starting point of the first iteration and separated from its nearest vertex by more than a given threshold. We have tested five methods to obtain this starting point for the second and next iterations:
   (a) Use the second point of the previous polygonal approximation as starting point in the second and next iterations ($\Delta sp = 1$).
   (b) Use the third point of the previous polygonal approximation as starting point in the second and next iterations ($\Delta sp = 2$).
   (c) Use the fourth point of the previous polygonal approximation as starting point in the second and next iterations ($\Delta sp = 3$).
   (d) Use the M / 2-th point of the previous polygonal approximation as starting point in the second and next iterations ($\Delta sp = \text{opposite}$). This proposal is similar to Horng's method [9].
   (e) Use the $M/(2^{i-1})$-th point of the previous polygonal approximation as starting point in the i-th iteration, similar to the binary search ($\Delta sp = \text{binary} - \text{opposite}$).
4. To obtain the polygonal approximation in the second and next iterations, taking into account the selection of the starting point according to the previous step, improved Salotti's method is used. In this case, the value of *ISE* corresponding to the previous polygonal approximation is used as value of pruning in the second and next iterations, so the computation time is greatly reduced.

3.4 Pikaz–Salotti proposal (*PHS*)

This proposal differs from the *PS* proposal only in the first iteration.

1. Obtain the polygonal approximation using the Pikaz's heuristic method [12] in the first iteration.
2. Obtain the polygonal approximation in the second iteration with the improved Salotti's method using:
   – The value of *ISE* corresponding to the first iteration as value of pruning.
   – The point of maximum associated error in the polygonal approximation of the first iteration as starting point.
3. The remaining iterations are similar to the second and next iterations for the *PS* proposal, where five methods are tested to select the starting point.

3.5 Advantages of *PS* and *PHS* proposals

These proposals have the next advantages:

– The Perez's method [8] and Pikaz's method [12] are faster than improved Salotti's method when a starting point is prefixed and no value of pruning is used.
– From the second iteration and the next iterations, the lowest value obtained of *ISE* in the previous iterations is used as value of pruning for the next iterations. Thus, the computation time is highly reduced. For this reason, the improved Salotti's method is used in these iterations.
– The value of *ISE* in each iteration is always less than or equal to the value of *ISE* of previous iterations. Due to this, the global optimal polygonal approximation is quickly reached.

4 Experimental results and discussion

The experiments have been carried out in a generic computer with processor Intel(R) Core(TM) i7-3930K CPU @ 3.20 GHz, 16 GB of RAM memory. The proposals *PS* and *PHS* have been tested using two groups of contours:

– 24 digital contours used in other works by the authors [16]. The number of points of these contours ranges from 554 to 2041. For the five methods to select the next starting point, all the global optimal polygonal approximations, between 5 and 50 points, have been calculated. From this group, 5520 global optimal polygonal approximations have been obtained. Figure 1 shows some of the digital contours belonging to the first group and their global optimal polygonal approximations for 30 points.
– 1400 digital contours from MPEG7-CE-Shape1 database [17]. The number of points of these contours

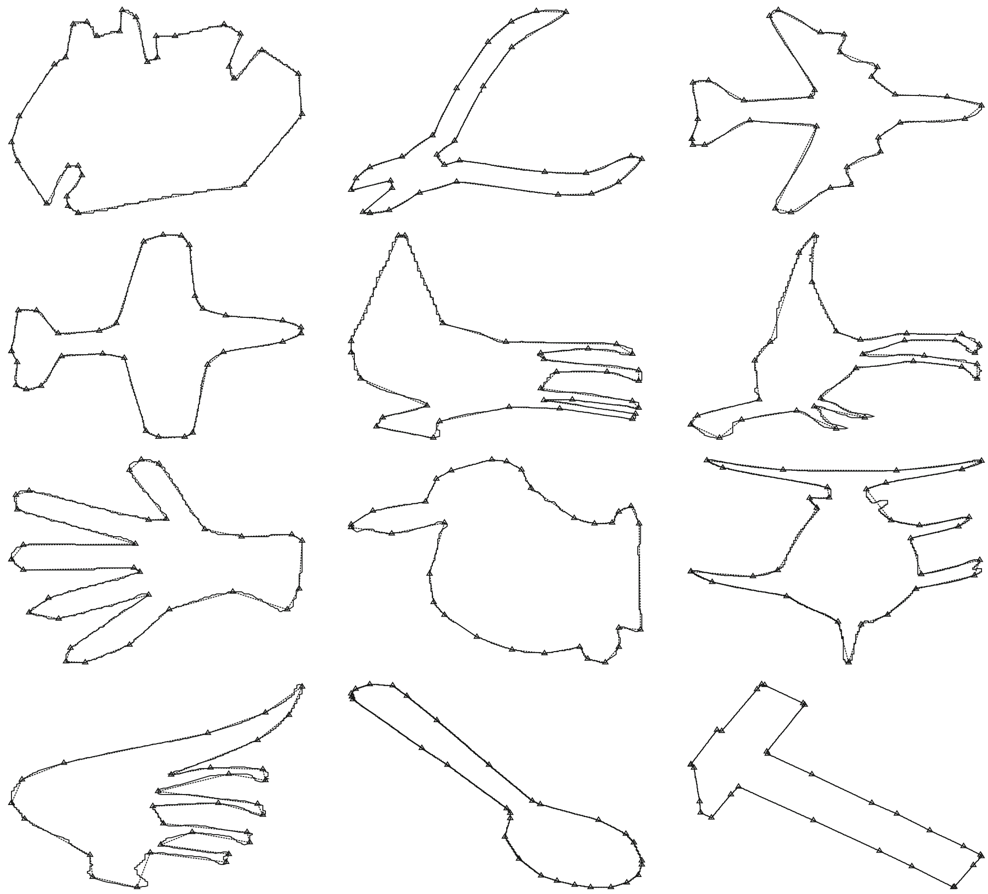


Fig. 1 Some of the digital contours used in this experiment and their global optimal polygonal approximations for 30 points

ranges from 146 to 5149. For the five methods to select the next starting point, all the global optimal polygonal approximations between 5 and 50 points have been calculated. From this group, 322460 global optimal polygonal approximations have been obtained.

In order to assess the proposed method and taking into account 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 similar to the original contours, with appropriate number of points (they are neither too high nor too low).

In order to test the accuracy of the proposed method, the global optimal polygonal approximation has been obtained using improved Salotti's method for all possible starting points. As it was said above, the results obtained from optimal approximations between 5 and 15 points and optimal approximations between 16 and 50 have been analyzed separately, in order to assess the proposals *PS* and *PHS* in polygonal approximations with a low number of points.

Taking into account the two groups of contours and the two proposals, two experiments have been performed.

4.1 First experiment

In this experiment the *PS* and *PHS* proposals, with the first group of contours, have been tested.

Table 1 shows the percentage of global optimal polygonal approximations obtained for the *PS* proposal in the first group of contours depending on the number of points, the method used to obtain the starting point and the number of iterations (*It*).

The results obtained for *PS* proposal for the first group of contours with $\Delta sp =$ opposite with four iterations were shown in [16].

Table 1 shows that for polygonal approximations with few points, if we use a number of iterations >3, the global optimal polygonal approximation is obtained in more than 90 % of cases. In this case, the best method is $\Delta sp = 2$.

Table 1 shows that for polygonal approximations with many points, if we use a number of iterations >2 (except in $\Delta sp = 1$), the global optimal polygonal approximation is

Table 1 Summary of the percentage of global optimal polygonal approximations obtained for the *PS* proposal for the first group of contours depending on the number of points, the method used to obtain the starting point and the number of iterations (*It*)

M	Method	$It \leq 2$	$It \leq 3$	$It \leq 4$	$It \leq 5$	$It \leq 6$	$It \leq 7$	$It \leq 8$	$It \leq 9$	$It \leq 10$
5–15	$\Delta sp = 1$	30.68	78.41	89.77	94.32	97.73	98.48	99.24	99.62	100.00
	$\Delta sp = 2$	57.20	90.15	96.21	96.97	98.86	99.62	100.00	100.00	100.00
	$\Delta sp = 3$	71.59	82.95	90.15	97.73	97.73	98.48	99.24	99.24	100.00
	$\Delta sp =$ opposite	76.45	85.98	95.08	97.35	98.11	98.86	99.24	99.62	100.00
	$\Delta sp =$ binary − opposite	76.45	91.32	95.87	96.69	97.11	97.52	98.35	99.59	100.00
16–50	$\Delta sp = 1$	60.60	85.95	93.10	97.50	99.29	99.64	99.88	100.00	100.00
	$\Delta sp = 2$	75.71	95.71	99.64	100.00	100.00	100.00	100.00	100.00	100.00
	$\Delta sp = 3$	88.57	99.52	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	$\Delta sp =$ opposite	95.45	97.50	98.69	99.29	99.76	100.00	100.00	100.00	100.00
	$\Delta sp =$ binary − opposite	95.45	99.87	100.00	100.00	100.00	100.00	100.00	100.00	100.00

Table 2 Summary of the percentage of global optimal polygonal approximations obtained for the *PHS* proposal for the first group of contours depending on the number of points, the method used to obtain the starting point and the number of iterations (*It*)

M	Method	$It \leq 2$	$It \leq 3$	$It \leq 4$	$It \leq 5$	$It \leq 6$	$It \leq 7$	$It \leq 8$	$It \leq 9$	$It \leq 10$	$It \leq 11$
5–15	$\Delta sp = 1$	17.80	38.64	56.82	74.24	85.23	90.91	94.32	97.73	99.24	100.00
	$\Delta sp = 2$	19.32	47.73	67.42	82.20	90.91	94.32	96.97	98.48	99.24	100.00
	$\Delta sp = 3$	19.70	49.62	69.32	85.23	93.94	95.08	97.35	98.11	98.86	100.00
	$\Delta sp =$ opposite	19.32	38.26	65.15	81.82	87.88	93.18	95.83	97.35	98.86	100.00
	$\Delta sp =$ binary − opposite	22.73	57.85	75.62	81.82	84.71	88.43	92.56	93.80	96.97	100.00
16–50	$\Delta sp = 1$	46.31	66.67	83.21	89.29	93.57	96.07	98.21	99.17	99.52	100.00
	$\Delta sp = 2$	41.55	68.21	81.43	88.93	93.93	96.79	97.98	98.81	99.88	100.00
	$\Delta sp = 3$	40.83	65.24	80.60	88.81	92.98	95.83	98.69	99.52	99.64	100.00
	$\Delta sp =$ opposite	42.74	62.50	85.95	91.90	94.52	96.55	98.69	99.17	99.40	100.00
	$\Delta sp =$ binary − opposite	42.34	80.39	89.74	94.03	97.92	99.09	99.35	99.48	99.52	100.00

obtained in more than 95 % of cases. In this case, the best method is $\Delta sp = 3$.

Horng's method is equivalent to $\Delta sp =$ opposite with *Iterations* ≤ 2. Table 1 shows that for this method the global optimal polygonal approximations are obtained in the 76.45 % of cases for approximations with few points and 95.45 % of cases in approximations with many points.

Table 2 shows the percentage of global optimal polygonal approximations obtained for the *PHS* proposal for the first group of contours depending on the number of points, the method used to obtain the first point and the number of iterations.

Table 2 shows that for polygonal approximations with few points, if we use a number of iterations >7 (except in binary-opposite), the global optimal polygonal approximation is obtained in more than 90 % of cases. In this case, the best method is $\Delta sp = 3$.

Table 2 shows that for polygonal approximations with many points, if we use a number of iterations >6, the global optimal polygonal approximation is obtained in more than 95 % of cases. In this case, the best method is $\Delta sp = 3$.

4.2 Second experiment

In this experiment the *PS* and *PHS* proposals, with MPEG7-CE-Shape1 database, have been tested.

Table 3 shows the percentage of global optimal polygonal approximations obtained for the *PS* proposal for MPEG7-CE-Shape1 database depending on the number of points, the method used to obtain the starting point and the number of iterations.

Table 3 shows that for polygonal approximations with few points, if we use a number of iterations >3, the global optimal polygonal approximation is obtained in more than 90 % of cases. In this case, the best method is $\Delta sp = 2$.

Table 3 shows that for polygonal approximations with many points, if we use a number of iterations >2 (except in $\Delta sp = 1$), the global optimal polygonal approximation is obtained in more than 95 % of cases. In this case, the best method is $\Delta sp = 3$.

Horng's method is equivalent to $\Delta sp =$ opposite with *Iterations* ≤ 2. Table 3 shows that for this method the global optimal polygonal approximations are obtained in

Table 3 Summary of the percentage of global optimal polygonal approximations obtained for the *PS* proposal for the MPEG7-CE-Shape1 database depending on the number of points, the method used to obtain the starting point and the number of iterations (*It*)

M	Method	$It \leq 2$	$It \leq 3$	$It \leq 4$	$It \leq 5$	$It \leq 6$	$It \leq 7$	$It \leq 8$	$It \leq 9$	$It \leq 10$	$It \leq 11$
5–15	$\Delta sp = 1$	59.71	83.47	91.82	95.26	97.28	98.67	99.27	99.53	99.81	100.00
	$\Delta sp = 2$	75.20	91.25	96.09	97.84	99.11	99.34	99.64	99.72	99.94	100.00
	$\Delta sp = 3$	81.81	90.21	94.60	98.09	98.92	99.14	99.62	99.71	99.86	100.00
	$\Delta sp =$ opposite	83.71	90.17	95.57	97.39	98.25	99.14	99.41	99.68	99.88	100.00
	$\Delta sp =$ binary − opposite	83.87	93.36	95.95	96.71	97.07	97.62	98.42	98.92	99.64	100.00
16–50	$\Delta sp = 1$	73.91	92.05	96.79	98.56	99.30	99.66	99.85	99.91	100.00	100.00
	$\Delta sp = 2$	86.97	98.14	99.50	99.82	99.92	99.97	99.98	99.99	100.00	100.00
	$\Delta sp = 3$	93.17	99.25	99.82	99.95	99.97	99.99	100.00	100.00	100.00	100.00
	$\Delta sp =$ opposite	97.35	98.39	99.45	99.63	99.78	99.85	99.90	99.95	100.00	100.00
	$\Delta sp =$ binary − opposite	97.33	99.61	99.79	99.84	99.88	99.88	99.91	99.92	100.00	100.00

Table 4 Summary of the percentage of global optimal polygonal approximations obtained for the *PHS* proposal for the MPEG7-CE-Shape1 database depending on the number of points, the method used to obtain the starting point and the number of iterations (*It*)

M	Method	$It \leq 2$	$It \leq 3$	$It \leq 4$	$It \leq 5$	$It \leq 6$	$It \leq 7$	$It \leq 8$	$It \leq 9$	$It \leq 10$	$It \leq 11$
5–15	$\Delta sp = 1$	26.95	49.16	67.99	78.90	85.59	90.68	93.96	96.21	98.05	100.00
	$\Delta sp = 2$	28.42	56.68	74.06	85.16	92.56	95.14	97.77	98.62	99.64	100.00
	$\Delta sp = 3$	28.79	54.40	72.11	85.80	92.53	94.31	97.10	98.19	98.91	100.00
	$\Delta sp =$ opposite	27.04	47.34	71.49	85.40	90.31	94.40	96.83	97.81	99.47	100.00
	$\Delta sp =$ binary − opposite	24.97	51.12	64.65	74.87	78.07	85.43	89.50	94.25	97.40	100.00
16–50	$\Delta sp = 1$	56.37	74.91	85.97	92.16	95.49	97.49	98.61	99.27	99.71	100.00
	$\Delta sp = 2$	56.50	78.02	89.67	94.83	97.44	98.67	99.37	99.63	99.79	100.00
	$\Delta sp = 3$	56.36	79.47	90.97	96.03	97.91	98.89	99.40	99.71	99.89	100.00
	$\Delta sp =$ opposite	54.16	70.18	88.30	94.10	96.03	97.64	98.72	99.27	99.66	100.00
	$\Delta sp =$ binary − opposite	48.81	70.53	77.48	80.75	84.77	87.83	91.60	95.48	98.03	100.00

the 83.71 % of cases in approximations with few points and 97.35 % of cases in approximations with many points.

Table 4 shows the percentage of global optimal polygonal approximations obtained by the *PHS* proposal for MPEG7-CE-Shape1 database depending on the number of points, the method used to obtain the starting point and the number of iterations.

Table 4 shows that for polygonal approximations with few points, if we use a number of iterations >8, the global optimal polygonal approximation is obtained in more than 95 % of cases, except in $\Delta sp =$ binary − opposite. In this case, the best method is $\Delta sp = 2$.

Table 4 shows that for polygonal approximations with many points, if we use a number of iterations >5, except in $\Delta sp =$ binary − opposite, the global optimal polygonal approximation is obtained in more than 90 % of cases. In this case, the best method is $\Delta sp = 3$.

In order to obtain a quantitative evaluation of the *PHS* proposal (which is the best one, as it will be shown below), a ratio of *ISE* corresponding to the different iterations has been calculated. The ratio of *ISE* is computed as:

$$\text{ratio} = 100 \times \frac{\text{ISE}_{it} - \text{ISE}_{opt}}{\text{ISE}_{it}} \quad (2)$$

where:

- ISE_{it} is the value of *ISE* corresponding to *it*-th iteration
- ISE_{opt} is the optimal value of *ISE*.

Table 5 shows that for polygonal approximations with few points, if we use three iterations, the ratio of *ISE* is <0.6 %. In the case of polygonal approximations with many points, if we use three iterations, the ratio of *ISE* is <0.01 %, except in $\Delta sp = 1$.

Finally, using the results obtained by the *PHS* proposal for the MPEG7-CE-Shape1, it has been studied how the length and the shape of the contour impact on the required number of iterations to determine the optimal result. Table 6 shows the means of the number of iterations

Table 5 Summary of the ratio of ISE ($100 \times (\mathrm{ISE}_{it} - \mathrm{ISE}_{opt})/\mathrm{ISE}_{it}$) in all iterations for the global optimal polygonal approximations obtained for the *PHS* proposal for the MPEG7-CE-Shape1 database depending on the number of points, the method used to obtain the starting point and the number of iterations (*It*)

M	Method	$It = 1$	$It = 2$	$It = 3$	$It = 4$	$It = 5$	$It = 6$	$It = 7$	$It = 8$	$It = 9$	$It = 10$	$It = 11$
5–15	$\Delta sp = 1$	6.770	2.389	0.566	0.247	0.127	0.088	0.059	0.022	0.013	0.001	0.0
	$\Delta sp = 2$	6.743	1.031	0.560	0.153	0.135	0.065	0.062	0.053	0.039	0.001	0.0
	$\Delta sp = 3$	6.876	1.032	0.582	0.412	0.102	0.037	0.016	0.013	0.008	0.004	0.0
	$\Delta sp =$ opposite	6.879	0.782	0.588	0.177	0.142	0.090	0.025	0.003	0.001	0.001	0.0
	$\Delta sp =$ binary − o.	6.828	0.730	0.227	0.127	0.108	0.091	0.088	0.030	0.005	0.002	0.0
16–50	$\Delta sp = 1$	1.626	0.491	0.150	0.022	0.008	0.004	0.002	0.001	0.0002	0.0	0.0
	$\Delta sp = 2$	1.647	0.137	0.008	0.004	0.001	$<10^{-4}$	$<10^{-5}$	$<10^{-5}$	$<10^{-6}$	0.0	0.0
	$\Delta sp = 3$	1.633	0.164	0.005	0.001	$<10^{-4}$	$<10^{-5}$	$<10^{-6}$	0.0	0.0	0.0	0.0
	$\Delta sp =$ opposite	1.633	0.031	0.002	0.001	0.001	$<10^{-4}$	$<10^{-4}$	$<10^{-5}$	$<10^{-6}$	0.0	0.0
	$\Delta sp =$ binary − o.	1.633	0.031	0.002	0.001	$<10^{-4}$	$<10^{-5}$	$<10^{-5}$	$<10^{-5}$	$<10^{-6}$	0.0	0.0

Table 6 Summary of means of the number of iterations necessary to obtain the global optimal polygonal approximations for the *PHS* proposal for the MPEG7-CE-Shape1 database depending on the number of points of the contour (*N*) and the number of points of the polygonal approximation (*M*)

N	$M = 10$	$M = 20$	$M = 30$	$M = 40$	$M = 50$
≤1000	3.328	2.528	2.168	1.922	1.564
1001–1500	3.407	2.912	2.671	2.185	1.967
1501–2000	3.352	2.529	2.666	2.392	2.147
2001–3000	3.833	3.000	2.500	2.229	1.979
≥300	3.384	2.769	3.115	2.346	2.538
All values	3.375	2.645	2.376	2.058	1.763

Table 7 Mean values of the number of iterations necessary to obtain the global optimal polygonal approximations for the *PHS* proposal for some similar categories of the MPEG7-CE-Shape1

Category	Deer/horse	Bone/hammer	Apple/hat	Frog/dog
Mean	1.847/4.079	2.493/1.604	2.883/1.810	1.719/4.195

necessary to obtain the global optimal polygonal approximations depending on the number of points of the contour (*N*) and the number of points of the polygonal approximation (*M*). The results show that:

– The number of iterations depends on *M*. When *M* is increased, the number of iterations decreases.
– The number of iterations does not depend on *N*.

To evaluate the impact of the shape of the contour, Table 7 shows the means of the number of iterations necessary to obtain the global optimal polygonal approximations in some similar categories of the MPEMPEG7-CE-Shape1. These categories are shown in Fig. 2. Taking into account the results, the number of iterations does not depend on the complexity of the shape because contours with similar complexity can need very different number of iterations.

4.3 Computational time

In order to evaluate the computational time of each proposal with the different methods to select the starting points, the mean values of time to obtain the global optimal polygonal approximations have been calculated for each value of *M* for MPEG7-database. The obtained results are shown in Figs. 3, 4, 5, 6 and 7.

These figures show that the obtained mean values of time for *PHS* proposal are much lower than the values obtained for *PS* proposal. The method used in the first iteration is the reason for this great difference. While Pikaz's heuristic method is $O(n \log n)$, Perez's method is $O(MN^2)$. For the next iterations, the computational time is similar for the two proposals. Moreover, the computation time is gradually decreased from the second iteration (in *PS* proposal) and from the third iteration (in *PHS* proposal) because the best value of *ISE* of the previous iterations is used as value of pruning.

This experiment shows again that the *PHS* proposal is the best proposal. Although in the *PHS* proposal are necessary more iterations to obtain the global optimal polygonal approximation, the computational time is much lower. The *PHS* proposal needs more iterations because it uses a heuristic method in the first iteration, and therefore, it produces a worse starting point for the second iteration. Taking into account the computation time the best methods to select the starting point are $\Delta sp = 2$ and $\Delta sp = 3$.

Finally, the computational time obtained by the *PHS* proposal using $\Delta sp = 3$ has been compared with:

– *PHS* proposal using original Salotti's method [15] in all iterations instead of improved Salotti's method.

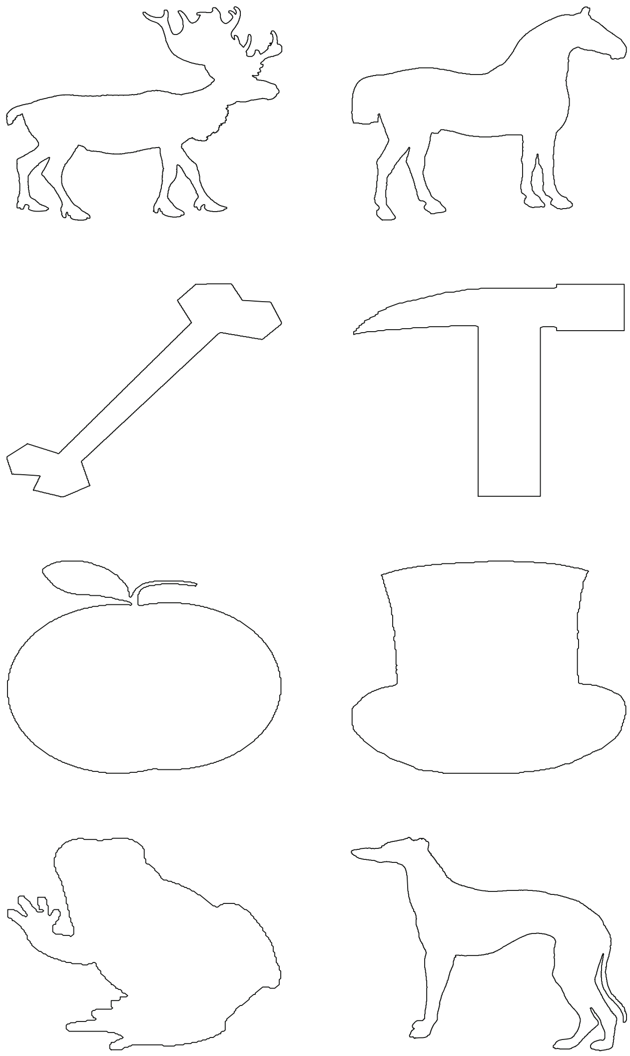


Fig. 2 Contours of the categories used to obtain the results of Table 7

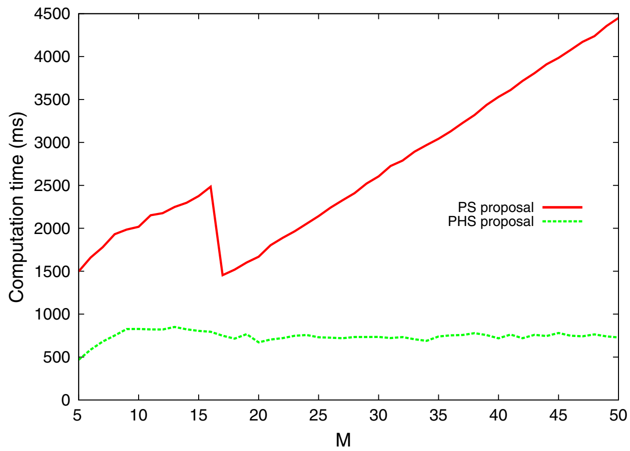


Fig. 3 Mean values of time for *PS* and *PHS* proposals for $\Delta sp = 1$

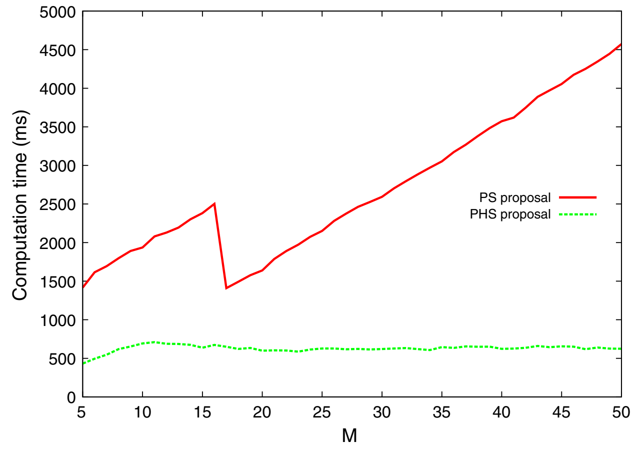


Fig. 4 Mean values of time for *PS* and *PHS* proposals for $\Delta sp = 2$

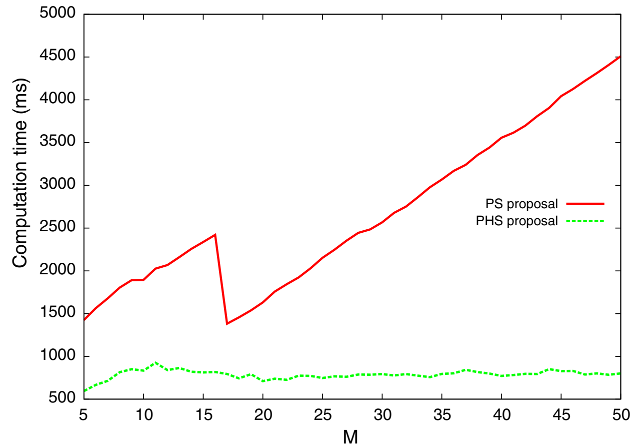


Fig. 5 Mean values of time for *PS* and *PHS* proposals for $\Delta sp = 3$

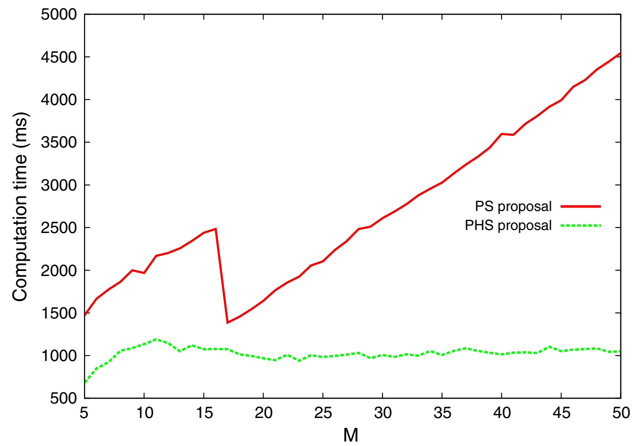


Fig. 6 Mean values of time for *PS* and *PHS* proposals for Δsp = opposite

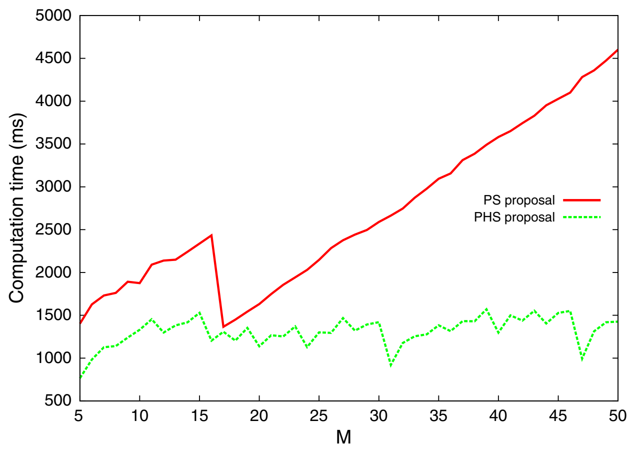


Fig. 7 Mean values of time for *PS* and *PHS* proposals for $\Delta sp = \text{binary} - \text{opposite}$

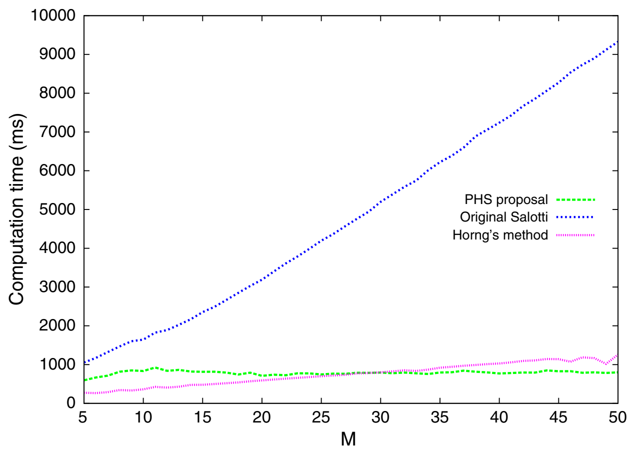


Fig. 8 Mean values of time for *PHS* proposal for $\Delta sp = 3$, *PHS* using original Salotti's method in all iterations and Horng's method

- Horng's method [9].

Figure 8 shows that:

- When the original Salotti's is used in the *PHS* proposal, the computational time is greater than the computational time when the improved Salotti's method is used.
- The computational time of Horng's method is better than the computational time obtained by *PHS* proposal for polygonal approximations with a number of points fewer than 29, and worst when the number of points is $\geq$29. However, Horng's method does not guarantee the optimal solution.

4.4 Discussion

Taking into account that proposal *PHS* is the best, and MPEG7-CE-Shape1 database contains a sufficient and representative number of contours, the following features can be highlighted:

- By using 11 iterations, the global optimal polygonal approximation will be obtained in all cases.
- For polygonal approximations with few points, the results are similar and very good in all cases when the number of iterations is >8.
- For polygonal approximations with many points, the results are similar and very good in all cases when the number of iterations is >7.

Though the global optimal polygonal approximation can be obtained in all cases using 11 iterations for any of the five methods to select the starting point, it can also be obtained with high probability using all methods with a low number of iterations and selecting the best polygonal approximation of the five methods. Let $P(e_{i,n})$ be the probability of not getting the global optimal polygonal approximation using the i-th method with n iterations, the probability P(E) of not obtain the global optimal polygonal approximation in n iterations if m methods to select the starting point are used is:

$$P(E) = \prod_{i=1,m} P(e_{i,n})$$

Therefore, the probability of obtaining the global optimal polygonal approximation is $1 - P(E)$. The values of $P(e_{i,n})$ for MPEG7-CE-Shape1 database can be obtained from Table 4. Let $v_{i,n}$ be the value of table for the i-th method with n iterations, then:

$$P(e_{i,n}) = 1 - \frac{v_{i,n}}{100.0}$$

Table 8 shows the values of $P(E)$ for MPEG7-CE-Shape1 database when the five methods are used. In this case:

- For polygonal approximations with few points, using 4 iterations, the probability of obtaining the global optimal polygonal approximation is >0.99.
- For polygonal approximations with many points, using 3 iterations, the probability of obtaining the global optimal polygonal approximation is >0.99.

Table 9 shows the values of $P(E)$ for MPEG7-CE-Shape1 database when the three best methods ($\Delta sp = 2, \Delta sp = 3$ and $\Delta sp = \text{opposite}$) are used. In this case:

- For polygonal approximations with few points, using 5 iterations, the probability of getting the global optimal polygonal approximation is >0.99.
- For polygonal approximations with many points, using 3 iterations, the probability of getting the global optimal polygonal approximation is >0.99.

Table 8 Values of $P(E)$ for MPEG7-CE-Shape1 database when all methods of obtaining the starting point are used

M	$It \leq 2$	$It \leq 3$	$It \leq 4$	$It \leq 5$	$It \leq 6$	$It \leq 7$	$It \leq 8$	$It \leq 9$	$It \leq 10$	$It \leq 11$
5–15	0.20	0.03	2.3×10^{-3}	1.6×10^{-4}	1.7×10^{-5}	2.1×10^{-6}	1.3×10^{-7}	1.2×10^{-8}	1.0×10^{-9}	0.0
16–50	0.02	0.001	3.4×10^{-5}	1.8×10^{-6}	1.4×10^{-7}	1.1×10^{-8}	6.0×10^{-10}	$< 10^{-10}$	$< 10^{-10}$	0.0

Table 9 Values of $P(E)$ for MPEG7-CE-Shape1 database when the three best methods of obtaining the starting point are used

M	$It \leq 2$	$It \leq 3$	$It \leq 4$	$It \leq 5$	$It \leq 6$	$It \leq 7$	$It \leq 8$	$It \leq 9$	$It \leq 10$	$It \leq 11$
5–15	0.37	0.10	0.02	3.1×10^{-3}	5.4×10^{-4}	1.5×10^{-4}	2.1×10^{-5}	5.4×10^{-6}	2.1×10^{-7}	0.0
16–50	0.08	0.01	1.1×10^{-3}	1.2×10^{-4}	2.1×10^{-5}	3.5×10^{-6}	4.8×10^{-7}	7.7×10^{-8}	7.7×10^{-9}	0.0

In these cases, when all methods or three methods are tested, the computation time is the one corresponding to the worse method, as parallel computing can be used.

5 Conclusions

The conclusions of this work can be summarized as follows:

– Two proposals (*PS* and *PHS*) to obtain global optimal polygonal approximations have been tested.
– Five methods to obtain the best starting point for the next iteration in the two proposals have been tested.
– The experimental results show that:

1. The *PHS* proposal, based on the Pikaz's method and the improved Salotti's method, is the best because its computational time is much lower.
2. By using 11 iterations in the *PHS* proposal, the global polygonal approximation is always obtained for all the methods used to obtain the starting point.
3. Horng's method obtains the global optimal polygonal approximation in the 83.71 % of cases for polygonal approximations with few points.
4. Horng's method obtains the global optimal polygonal approximation in the 97.35 % of cases for polygonal approximations with many points.
5. For polygonal approximations with many points, if the three best methods ($\Delta sp = 2, \Delta sp = 3$ and $\Delta sp =$ opposite) with three iterations are used, and the best of them is selected, the probability of obtaining the global optimal polygonal approximation is >0.99.
6. For polygonal approximations with few points, if the three best methods ($\Delta sp = 2, \Delta sp = 3$ and $\Delta sp =$ opposite) with five iterations are used, and the best of them is selected, the probability of obtaining the global optimal polygonal approximation is >0.99.
7. In most cases, the best method to select the starting point is $\Delta sp = 3$, although any method can achieve the global optimal polygonal approximation.
8. In order to obtain the global optimal polygonal approximation with few points, more iterations are necessary but the optimum is achieved as well.

– Since the computational complexity of the Salotti's method is close to $O(N^2)$ [10] and Pikaz's method is $O(n \log n)$ [12], it can be considered that the computational complexity of the proposed method is also close to $O(N^2)$ and between $O(N^2) - O(MN^2)$.

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

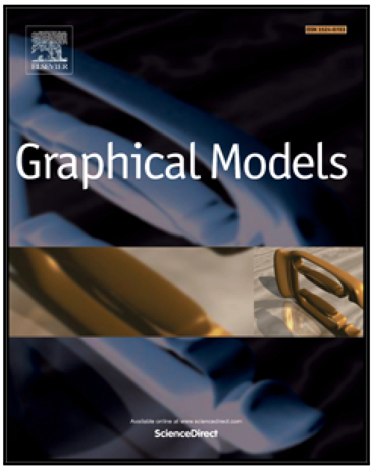
References

1. Lowe DG (1987) Three-dimensional object recognition from single two-dimensional images. Artif Intell 31:355–395
2. Grauman K, Darrell T (2004) Fast contour matching using approximate earth mover's distance. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004, vol 1, pp I–220–I–227. doi:10.1109/CVPR.2004.1315035
3. Chen M, Xu M, Franti P (2012) A fast multiresolution polygonal approximation algorithm for GPS trajectory simplification. IEEE Trans Image Process 21(5):2770–2785. doi:10.1109/TIP.2012.2186146 (ISSN 1057-7149)
4. Douglas DH, Peucker TK (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartogr Int J Geogr Inf Geovis 10(2):112–122. doi:10.3138/FM57-6770-U75U-7727
5. Chan WS, Chin F (1996) Approximation of polygonal curves with minimum number of line segments or minimum error. Int J Comput Geom Appl 06(01):59–77. doi:10.1142/S0218195996000058 (ISSN 0218-1959)
6. Schuster GM, Katsaggelos AK (1998) An optimal polygonal boundary encoding scheme in the rate distortion sense. IEEE Trans Image Process 7:13–26
7. Chen DZ, Daescu O (2003) Space-efficient algorithms for approximating polygonal curves in two-dimensional space. Int J

Comput Geom Appl 13(02):95–111. doi:10.1142/S0218195903001086 (ISSN 0218-1959)
8. Perez J-C, Vidal E (1994) Optimum polygonal approximation of digitized curves. Pattern Recogn Lett 15(8):743–750. doi:10.1016/0167-8655(94)90002-7 (ISSN 0167-8655)
9. Horng J-H, Li JT (2002) An automatic and efficient dynamic programming algorithm for polygonal approximation of digital curves. Pattern Recogn Lett 23(1–3):171–182. doi:10.1016/S0167-8655(01)00098-8 (ISSN 0167-8655)
10. Salotti M (2001) An efficient algorithm for the optimal polygonal approximation of digitized curves. Pattern Recogn Lett 2(22): 215–221. doi:10.1016/S0167-8655(00)00088-X (ISSN 0167-8655)
11. Kolesnikov A, Franti P (2007) Polygonal approximation of closed discrete curves. Pattern Recogn 40(4):1282–1293. doi:10.1016/j.patcog.2006.09.002 (ISSN 0031-3203)
12. Pikaz A, Dinstein I (1995) An algorithm for polygonal approximation based on iterative point elimination. Pattern Recogn Lett 16(6):557–563. doi:10.1016/0167-8655(95)80001-A (ISSN 0167-8655)
13. Pei S-C, Horng J-H (1996) Optimum approximation of digital planar curves using circular arcs. Pattern Recogn 29(3):383–388. doi:10.1016/0031-3203(95)00104-2 (ISSN 0031-3203)
14. Kolesnikov A, Franti P (2003) Reduced-search dynamic programming for approximation of polygonal curves. Pattern Recogn Lett 24(14):2243–2254. doi:10.1016/S0167-8655(03)00051-5 (ISSN 0167-8655)
15. Salotti M (2002) Optimal polygonal approximation of digitized curves using the sum of square deviations criterion. Pattern Recogn 35(2):435–443. doi:10.1016/S0031-3203(01)00051-6 (ISSN 0031-3203)
16. Carmona-Poyato A, Aguilera-Aguilera EJ, Madrid-Cuevas FJ, Lopez-Fernandez D (2015) New method for obtaining optimal polygonal approximations. In: Paredes R, Cardoso JS, Pardo XM (eds) Pattern recognition and image analysis, number 9117 in Lecture Notes in Computer Science. Springer, Berlin, pp 149–156 (**ISBN 978-3-319-19389-2 978-3-319-19390-8**)
17. Latecki LJ, Lakamper R, Eckhardt T (2000) Shape descriptors for non-rigid shapes with a single closed contour. In: Proceedings of IEEE conference on computer vision and pattern recognition, 2000. CVPR 2000, vol 1, pp 424–429. doi:10.1109/CVPR.2000.855850

Chapter 5

Fourth contribution: Fast computation of optimal polygonal approximations of digital planar closed curves



Fast computation of optimal polygonal approximations of digital planar closed curves

E.J. Aguilera-Aguilera, A. Carmona-Poyato, F.J. Madrid-Cuevas, M.J. Marín-Jiménez

PII: S1524-0703(16)00017-5
DOI: 10.1016/j.gmod.2016.01.004
Reference: YGMOD 943

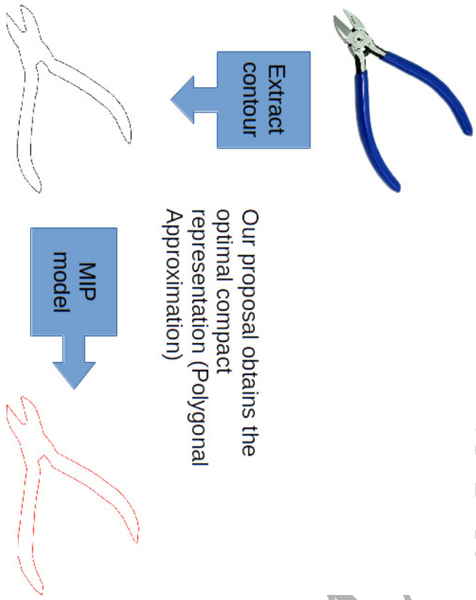
To appear in: *Graphical Models*

Received date: 27 July 2015
Revised date: 10 January 2016
Accepted date: 22 January 2016

Please cite this article as: E.J. Aguilera-Aguilera, A. Carmona-Poyato, F.J. Madrid-Cuevas, M.J. Marín-Jiménez, Fast computation of optimal polygonal approximations of digital planar closed curves, *Graphical Models* (2016), doi: 10.1016/j.gmod.2016.01.004

Highlights

- A novel method to solve the min-# polygonal approximation problem is proposed.
- The approach uses a modified Mixed Integer Programming model to solve the min-# problem.
- The proposed model is smaller than previous proposals.
- The novel procedure obtains the optimal solution faster than state-of-the-art methods.
- Only one execution of our procedure is needed to assure the optimality of the solution.


Extract contour
Our proposal obtains the optimal compact representation (Polygonal Approximation)
MIP model

Fast computation of optimal polygonal approximations of digital planar closed curves

E.J. Aguilera-Aguilera[a,*], A. Carmona-Poyato[a], F.J. Madrid-Cuevas[a], M.J. Marín-Jiménez[a]

[a] *Department of Computing and Numerical Analysis, Córdoba University. 14071-Córdoba. Spain*

Abstract

We face the problem of obtaining the optimal polygonal approximation of a digital planar curve. Given an ordered set of points on the Euclidean plane, an efficient method to obtain a polygonal approximation with the minimum number of segments, such that, the distortion error does not excess a threshold, is proposed. We present a novel algorithm to determine the optimal solution for the min-# polygonal approximation problem using the sum of square deviations criterion on closed curves.

Our proposal, which is based on Mixed Integer Programming, has been tested using a set of contours of real images, obtaining significant differences in the computation time needed in comparison to the state-of-the-art methods.

Keywords: Digital planar curves, Polygonal approximation, Integral Square Error, Mixed Integer Programming, Discrete optimization, min-# polygonal approximation problem

1. Introduction

Since Attneave [1] pointed out that the information is concentrated at dominant points, their detection has become an important research area in computer

*Corresponding author. Tel.: 34 957218630. Fax: 34 957218630

Email addresses: i22agage@uco.es (E.J. Aguilera-Aguilera), ma1capoa@uco.es (A. Carmona-Poyato), ma1macuf@uco.es (F.J. Madrid-Cuevas), mjmarin@uco.es (M.J. Marín-Jiménez)

vision. Dominant points are those that can describe the curve for visual perception and recognition. In the literature two major categories can be found: corner detection methods [2, 3, 4] and polygonal approximation methods [5, 6, 7].

In computer vision polygonal approximation of digital planar curves is an important task for a variety of applications like simplification on vectorization algorithms [8], image analysis [9], shape analysis [10], object recognition [11], Geographical Information Systems [12], and digital cartography [13].

The main idea behind polygonal approximations of digital planar curves is to provide a compact representation of the original shape with reduced memory requirements, preserving the important shape information.

The optimization problem of polygonal approximations has been formulated into two separated ways, depending on the objective function that we want to minimize:

- min-#: Minimize the number of line segments M that forms a polygonal approximation, such that, the distortion error does not excess a threshold ε. Moreover the optimal solution should have the lowest distortion associated among all the solutions with the same number of line segments.
- min-ε: Given a number of line segments M, minimize the distortion error associated to the polygonal approximation.

In the literature, several alternatives to solve the min-# and min-ε problem can be found, using heuristic, metaheuristic and optimal approaches. The selection of the distortion measure used in the algorithm is task dependent: the use of the L_∞-norm is used to assure that the maximum deviation does not exceed the threshold provided by the user, whereas, the use of the L_2-norm provides a polygonal approximation whose distortion (Integral Square Error) is lower than the provided threshold.

For instance, to solve the min-ε problem using the L_2-norm several metaheuristics have been applied: genetic algorithms [14], ant colony search algorithms [15], integer particle swarm optimization algorithms [7], etc.

The main problem of using the metaheuristic approaches is the computational cost. There are several proposals in the literature that use some heuristic with a low computational cost: split based methods [16], merge based methods [17, 18], merge-split based methods [5], etc.

This problem is solved optimally by using dynamic programming approach [19] and graph approach [6]. These algorithms obtain optimal solutions, however a high computational burden is required to obtain them. A more recent and faster method based on Mixed Integer Programming was proposed in [20].

In practice, the reduction of the description of a shape with a maximum tolerance error, that is, the min-# problem, is a more common task. Therefore, a great variety of algorithms have been proposed to solve this problem. For example, the min-# problem using the L_2-norm has been solved by using metaheuristic solutions: genetic algorithms [21], ant colony optimization [15], particle swarm optimization [22, 23], tabu search [24], etc. More rapid algorithms are based on other heuristics like a split approach [13], merge approach [25], graph approach [26] etc.

The min-# can also be solved optimally by using a modified version of the dynamic programming approach proposed by Perez and Vidal [19] and a graph approach by Salotti [27]. The original algorithm, by Perez and Vidal, was used to obtain the polygonal approximation with a fixed number of segments M which has the minimum distortion error associated (min-ε problem). This modified version of the algorithm was proposed in [27] and is shown in Algorithm 1. The idea is to increase the number of segments needed to reach the last point of the curve until the error associated to the polygonal approximation is lower than a threshold ε.

```
Data: c (Digital planar curve), ε (maximum error)
Result: The optimal polygonal approximation
var NP // Number of points;
var NS // Maximum number of segments;
var g // used to memorize the minimum global error to reach any point of
the contour using any number of segments;
var Points // Points of the digital planar curve;
var Father // Array that contains the ending point of the previous
segment;
g[1,0] ← 0;
for n ← 2 to NP do
    g[n,0] ← maxValue;
end
m ← 0;
repeat
    m ← m + 1;
    for n ← 2 to NP do
        // Search the minimum error to reach point n with m segments;
        g[n,m] ← min(g[i,m−1] + ISE(i,n));
        // Store the index i_min of the point with the minimum error;
        Father[n,m] ← i_min;
    end
until g[m,n] < ε;
```

Algorithm 1: Modified version of the algorithm by Perez and Vidal [19]

The main drawback of using optimal algorithms is the computational cost required to achieve the solution. Some improvements have been made to reduce this computational burden. For instance, Horng and Li [28] proposed an heuristic method to determine the initial point for the method based on Dynamic Programming used to solve the min-ε. This method uses two iterations of the Dynamic Programming method to obtain a polygonal approximation. Kolesnikov and Fränti [29] introduced a method to obtain polygonal approximations based on a cyclically extended closed curve of double size. The method selects the best starting point by searching on the extended search space for the extended curve. Both methods obtain good solutions using different heuristic approaches, however, neither of these methods can assure the optimality of the solution.

However, optimal polygonal approximations are very important because they are commonly used to assess the quality of the suboptimal polygonal approximations obtained by suboptimal methods.

The main contributions of this paper are: (i) a novel Mixed Integer Programming (MIP) model to solve the min-# polygonal approximation problem using a different approach to the state-of-the-art optimal algorithms, (ii) our proposal reduces significantly the computation time to obtain the optimal solution; (iii) the proposed MIP model is smaller than previous proposals.

Section 2 summarizes the most important measures to evaluate the quality of polygonal approximations that appear in the literature. In Section 3, we formulate the problem and present the MIP model that solves this problem. Section 4 defines the experiments carried out and the results obtained. In Section 5, we discuss the results obtained and the most important aspects of the proposed method. Finally, Section 6 presents the main conclusions of this paper.

2. Measures to assess the quality of polygonal approximations

The quality of a polygonal approximation is quantified by the amount of data reduction obtained and the closeness of the approximation to the original curve. Several authors have faced this problem using different approaches.

Sarkar [30] proposed a method to evaluate the quality of the polygonal approximation, based on the distortion associated to the solution (Integral Square Error) and the compression ratio obtained by the solution. This measure, called Figure of Merit (FOM), is defined as

$$\mathrm{FOM} = \frac{\mathrm{CR}}{\mathrm{ISE}} \tag{1}$$

where the compression ratio (CR) is defined as the number of points N of the original contour divided by the number of points M of the polygonal approximation ($\text{CR} = \frac{\text{N}}{\text{M}}$). The ISE is the sum of the squared orthogonal distances from the points of the original contour to the line of the polygonal approximation which approximates them.

Rosin [31] showed that the FOM measure is biased to favor approximations with larger number of line segments, because its two terms are unbalanced. Another problem of the FOM is that is not suitable to assess polygonal approximations with different number of points M. To solve these drawbacks, Rosin [31] proposed to use a novel measure. The new measure, named as Rosin's merit, is defined based on two components: fidelity and efficiency. Fidelity measures how well the suboptimal polygonal approximation fits the curve relative to the optimal polygon in terms of the approximation error. This component is defined as

$$\text{Fidelity} = \frac{E_{\text{opt}}}{E_{\text{approx}}} \tag{2}$$

where E_{opt} is the error associated to the optimal solution and E_{approx} is the error of the suboptimal polygonal approximation. Both, the optimal solution and the suboptimal polygonal approximation, use the same number of points M.

Efficiency measures how compact is the polygonal approximation relative to the optimal polygonal approximation that incurs in the same error. This measure is defined as

$$\text{Efficiency} = \frac{M_{\text{opt}}}{M_{\text{approx}}} \tag{3}$$

where M_{approx} is the number of points M of the polygonal approximation which is been tested, and M_{opt} is the number of points that an optimal solution needs to obtain the same error associated to the polygonal approximation which is been tested.

These two components are combined using a geometric mean that is named Rosin's merit. This measure is defined as

$$\text{Merit} = \sqrt{\text{Fidelity} \times \text{Efficiency}} \tag{4}$$

The main advantage of the Rosin's merit over the FOM is that it can be used to compare two polygonal approximation with different number of points. However, Carmona-Poyato et al. [32] showed that the Rosin's merit presents one significant disadvantage: this measure does not take into account if the number of points M of the polygonal approximation faithfully represents the original shape. That is, an optimal polygonal approximation with very low number of segments ($M = 3$, for instance) will obtain better merit than a suboptimal solution with a reasonable number of points. A modified version of the Rosin's merit was proposed in [32] to solve this drawback. The novel measure uses an optimal solution which is taken as reference, and for each optimal polygonal approximation i is obtained a value $M(i)$. Using these values, the proposed fidelity and efficiency are computed. Finally, Carmona's merit is computed using the geometric mean of these two components.

State-of-the-art techniques to assess polygonal approximations use optimal solutions to be computed. However, optimal algorithms require a high computational cost to obtain the optimal polygonal approximation. We face the problem of obtaining the optimal polygonal approximation with the minimum number of segments using the sum of square error measure on closed curves. Our proposal is based on the Mixed Integer Programming optimization framework.

3. MIP model formulation

3.1. A brief introduction to Mixed Integer Programming

Mixed Integer Programming (MIP) is an optimization technique broadly used to solve discrete optimization problems. Some examples of real problems are: allocation of distributed generators in radial distribution systems [33], data envelop analysis [34], modern semiconductor manufacturing systems [35] and augmented reality [36].

A MIP problem is the minimization or maximization of a linear objective function which is subject to a set of linear constraints. The objective function is defined as

$$z = \min \mathbf{c}^T \mathbf{x},\ c, x \in \mathbb{R}^n \tag{5}$$

where x is a vector of decision variables and c is the vector of cost values associated to the decision variables. This objective function is subject to a set of linear constraints which can be defined as

$$Ax \leq b \tag{6}$$

where A is called constraint matrix. Decision variables may take values between an upper and a lower bound which is defined as:

$$l \leq x \leq u \tag{7}$$

Some decision variables are required to take integer values. Integer variables that must take values 0 or 1 are called binary variables and play a special role in MIP modeling and solving.

The MIP problems are solved using two phases. In the first stage, the model is solved using the Simplex algorithm (introduced by Dantzig [37]) as if there were no integer restrictions. The Simplex algorithm obtains a value for the objective function that is used as a lower or upper bound, depending on whether we are maximizing or minimizing the objective function.

The next stage is solved using the algorithm known as Branch & Bound (introduced by Land and Doig [38]) or some variant of the algorithm (Branch & cut, Branch & price, etc). This algorithm relies on a process of searching in a tree. The method works by enumerating all the possible combinations of the integer variables in the tree. Each node of this tree is a continuous optimization problem, based on the original problem, but with some of the decision variables set to a fixed value of the integer range provided. The root of the tree is the relaxed version of the optimization problem without integrality constraints. The value of the objective function of the root node is always a bound of any integer solution found in the tree. The Branch & Bound method generates the tree by selecting an integer variable x_i and adds a node for each possible integer value of the range. The algorithm selects one of these new nodes of the tree, and solves the model using Simplex with this variable x_i fixed to the selected value. We can obtain four different results:

- The subproblem is infeasible if the Simplex algorithm returns that the subproblem is unbounded. Therefore, any further restriction of this subproblem would be also infeasible, then, this node should be pruned.
- The subproblem is feasible, however the objective function obtained is worse than a previous integer solution, therefore, no children of this subproblem could improve the objective function. This node should be pruned.
- The subproblem is feasible, all the integer restrictions are satisfied and the objective function is better than any previous known integer solution. Then, the method stores this integer solution as the best feasible solution found. This node is a leaf of the tree because no children nodes can be generated.
- If none of the above occurs, then some integer decision variable x_j is fractional at optimality. The method branches the problem on this variable by generating a children node of x_j for each integer value that the variable can take from its possible range. If there are several decision variables with fractional values, then the method selects one of these variables using some of the heuristic method proposed (e.g. [39, 40]).

This algorithm repeats until no new nodes can be generated, because no fractional decision variable can be used to branch.

3.2. Problem formulation

We can formulate the problem as follows. Let suppose a closed curve C defined by a number of ordered points N, where $N > 2$. A segment S can be determined using two points P_i and P_j belonging to the curve C.

A polygonal approximation A can be defined as an ordered number of segments M ($A = \{S_1, S_2, \cdots, S_M\}$), such that, the first point of segment S_{i+1} is the last point of segment S_i.

Any polygonal approximation A has an associated error. Several measures can be used to define the error associated, however we use the L_2-norm which is used to define the Integral Square Error. We can define this error measure associated to a segment S_{ij} formed by points P_i and P_j as

$$\Delta(i,j) = \sum_{k=i}^{j} \mathrm{dist}(P_k, S_{ij})^2 \tag{8}$$

where $\mathrm{dist}(P_k, S_{ij})$ is the orthogonal distance from the point P_k to the segment S_{ij} that approximates it. Then, Integral Square Error (ISE) is the sum of the distortion associated to each segment belonging to the polygonal approximation. We can define this measure as

$$\mathrm{ISE} = \sum_{k=1}^{M} \Delta(i_k, i_{k+1}) \tag{9}$$

where i_k and i_{k+1} are the indexes of the initial and end points of the segment S_k.

Therefore, given an error threshold ε the polygonal approximation ISE associated is less than or equal to this threshold. The solution provided must be optimal in the number of segments M and in the error ISE associated. That is, the polygonal approximation should have the minimum number of segments M whose distortion associated does not exceed the error threshold ε and the polygonal approximation should have associated the minimum error among the polygonal approximations with the same number M of segments.

3.3. MIP model proposed for closed curves

As explained above, the keypoint is to create a suitable MIP model which defines the problem. In this paper we propose a MIP model to solve the min-# polygonal approximation problem on closed curves. Our model defines a set of binary decision variables. We use these binary variables to define whether a segment $S_{i,j}$ is used in the polygonal approximation provided as the solution. For this reason, we define the following set of variables:

$$\forall x_{i,j} \in \{0, 1\} \, \forall i \in [1, 2, \cdots, N] \, \forall j \in [1, 2, \cdots, N], \, i \neq j \tag{10}$$

These decision variables are used to define all the possible solutions. Due to the fact that we are considering all possible solutions, the method does not need to fix any point as the initial point of the solution. The decision variables $x_{i,j}$, where $i = j$, are not defined in the model, because a valid segment is formed by two different points. This reduction of decision variables regarding the MIP model in [20] causes this model to be more compact. We have defined $(N-1) \times (N-1)$ decision variables instead of $N \times N$ defined in the model for solving the min-ε polygonal approximation problem.

Taking into account these decision variables, the objective function is defined as:

$$z = \min \sum_{i=1}^{N} \sum_{j=1}^{N} x_{i,j} \cdot (\varepsilon + 1) + \sum_{i=1}^{N} \sum_{j=1}^{N} x_{i,j} \cdot \Delta(i, j) \, , i \neq j \tag{11}$$

where, all the segments belonging to the final solution are summed in the first summation of the objective function while the second summation is the distortion (ISE) of the solution. The first summation of the objective function uses the threshold error ε (plus one) to penalize adding a new segment to the solution. The first summation of the objective function is always greater than the second summation, because the distortion of the solution should be lower than the error threshold ε. Therefore, the second summation of the objective function is defined to obtain the solution with the minimum distortion associated, among all the solutions with the minimum number of segments. A further explanation of the objective function is given in Section 3.4.

The error associated to the polygonal approximation should be lower than the threshold ε provided by the user. Therefore, to take into account this information we define a new constraint as

$$\sum_{i=1}^{N}\sum_{j=1}^{N} x_{i,j} \cdot \Delta(i,j) \leq \varepsilon \,, i \neq j \tag{12}$$

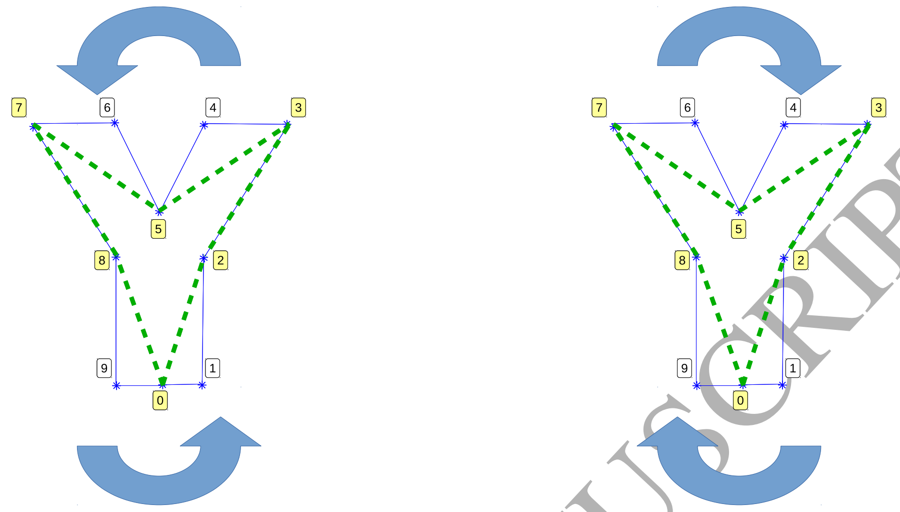
This model has also to take into account that we want to obtain closed polygonal approximations (not isolated segments). For this reason, the new set of constraints are defined as

$$\sum_{r=1}^{N} x_{i,r} = \sum_{c=1}^{N} x_{c,i} \, \forall i \in [1,2,3,\cdots,N] \,, i \neq r \,, i \neq c \tag{13}$$

A new problem arises taking into account how a solution is represented using the defined decision variables. A polygonal approximation is defined using $M-1$ segments $S_{i,j}$ where $i < j$ and one segment $S_{a,b}$ where $a > b$. For instance, the polygonal approximation which is shown in Fig. 1(a) is defined setting the decision variables $x_{0,2}, x_{2,3}, x_{3,5}, x_{5,7}, x_{7,8}, x_{8,0} = 1$ in the model. However, this polygonal approximation may be defined in the opposite direction ($x_{0,8}, x_{8,7}, x_{7,5}, x_{5,3}, x_{3,2}, x_{2,0} = 1$) as is shown in Fig 1(b). These two representations define the same solution. This problem is known in Mixed Integer Programming as symmetries [41], and makes the solver to take more computational time to obtain the optimal solution. Therefore, these symmetries must be avoided by mainly adding new constraints, which define that these symmetric solutions are not considered valid. To avoid this problem we have defined a new constraint as

$$\sum_{i=1}^{N}\sum_{j=1}^{i} x_{i,j} = 1 \,, i \neq j \tag{14}$$

where we state that only the solutions defined in counterclockwise direction



(a) Polygonal approximation defined in counterclockwise direction

(b) Polygonal approximation defined in clockwise direction

Figure 1: Polygonal approximation defined in counterclockwise (a) and clockwise (b) direction.

are valid. Therefore, this constraint expresses that a valid polygonal approximation must contain one segment $S_{a,b}$ where $a > b$.

3.4. Model analysis

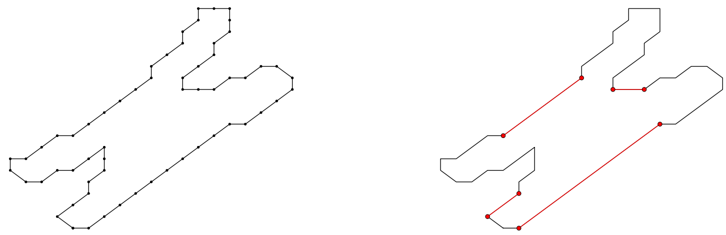
Our MIP model is defined using a set of linear constraints (see inequalities 12 to 14). These sets of constraints are used to define the problem and to obtain a feasible optimal solution for the min-# polygonal approximation problem. In this section we evaluate the convenience of using the defined constraints.

We analyze the objective function of the model first. The min-# problem definition states that we are seeking the polygonal approximation with the minimum number of segments which have an error associated that is lower than a threshold ε provided by the user. The objective function that is shown in equation 11 minimizes the summation of binary decision variables $x_{i,j}$, that represent the number of segments. However, we define that this number of segments are multiplied by a constant value ($\varepsilon + 1$). By multiplying the number of segments M by the value $\varepsilon + 1$, we are not modifying the optimal solution, but ordering the solutions according to the number of segments and the distortion associated. This is achieved because the distortion associated to the polygonal approximation should be lower or equal than the error threshold ε and the number of segments multiplied by $\varepsilon + 1$ will be greater than this distortion. For instance, two solutions a and b with the same number of segments $M_a = M_b$ have associated distortion errors such that $\mathrm{ISE_a} < \mathrm{ISE_b}$. Therefore, the objective functions obtained are $z_a = M_a \cdot (\varepsilon + 1) + \mathrm{ISE_a}$ and $z_b = M_b \cdot (\varepsilon + 1) + \mathrm{ISE_b}$. Since the number of segments are equal, then $M_a \cdot (\varepsilon + 1) = M_b \cdot (\varepsilon + 1)$. Therefore, the minimum value for the objective function is obtained for that polygonal approximation which has the minimum distortion associated, that is, $\mathrm{ISE_a}$.

Let us suppose that the constraint defined in inequality 12 is removed. This linear constraint defines that distortion associated to the solution is lower or equal than a threshold ε defined by the user. Without this constraint, the MIP solver always obtains a trivial solution with 3 points, which is the minimum polygonal approximation that we can define. This trivial solution does not meet the requirements of the min-# problem. Therefore, this constraint is mandatory to define a suitable model for this problem.

The constraint defined in inequality 13 forces the MIP solver to select consecutive line segments, that is, if the line segment consisting of points i and j is selected ($x_{i,j} = 1$) then some segment which ends with point i must be selected; and some segment which starts with point j must be selected as well. For instance, if the constraint 13 is not present in the model, we can define a solution using four line segments ($x_{2,11} = x_{21,23} = x_{38,43} = x_{58,60} = 1$) for the contour known as chromosome, however this solution is not a valid polygonal approximation as is shown in Fig. 2(b). Therefore, the constraint defined in inequality 13 is mandatory to obtain feasible optimal solutions.

To avoid the symmetric solutions, the constraint defined in inequality 14 is used. If this constraint is removed, the solver will consider the symmetric solutions valid, therefore the number of feasible solutions to explore grows



(a) Chromosome original contour (b) Isolated line segments

Figure 2: Original contour named as chromosome appear in Fig. 2(a) and the isolated line segments in Fig. 2(b).

exponentially. This constraint is not mandatory to obtain optimal feasible solutions, however, considerably reduces the computation time required for solving the min-# problem.

The defined MIP model is finally solved by using the Branch & Bound algorithm, introduced by Land and Doig [38]. This algorithm enumerates all possible solutions to the problem creating a search tree: the nodes are new problems with some decision variables equal to some value. The method searches for the optimal solution in the search tree, pruning those solutions which do not produce better results than the best solution found so far. To illustrate how the algorithm works, a little example is presented below.

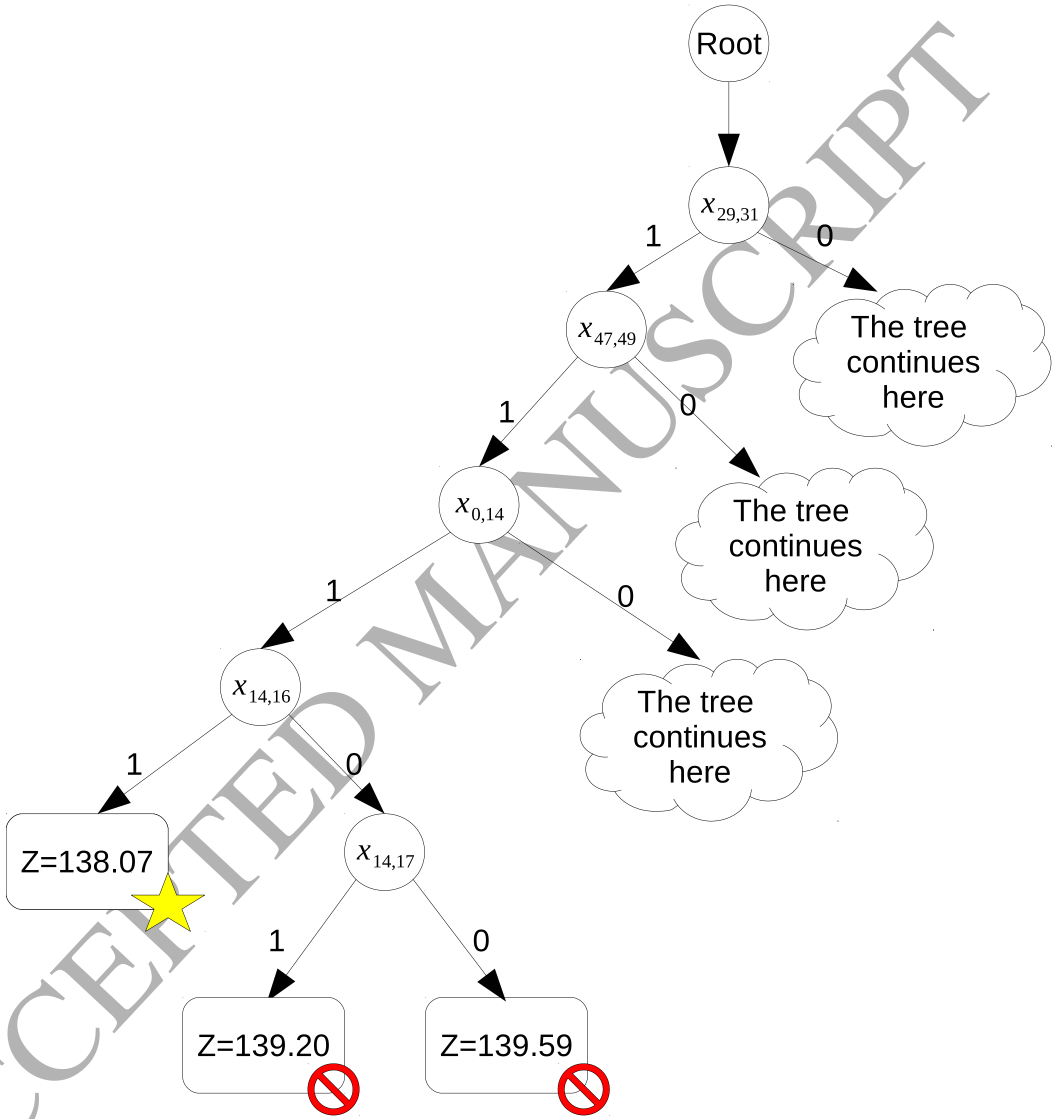
Let us suppose we want to solve the min-# problem for contour chromosome (Fig. 2(a)) with distortion threshold $\varepsilon = 12$. The algorithm first runs the Simplex method (introduced by Dantzig [37]) on the proposed model with no integer restrictions. This relaxed version of the model is the root node of our search tree. We obtain a value of the objective function of 122.94 with all the decision variables with an integer value except the decision variables $x_{29,31}, x_{29,47}, x_{31,37}, x_{37,47}$. Notice that the objective function for the relaxed version of the problem will be lower than the objective function of the solution where all the integrality constraints are satisfied. We should select a node (decision variable) to branch and then fix possible values for this variable. In the literature, several branching techniques can be found [42]. To simplify the explanation we select the first fractional (not integer) decision variable, i.e., the depth first strategy. We select the decision variable of $x_{29,31}$ and fix the value to 1. We obtain a value for objective function of $z = 129.95$ where all decision variables have taken integer values except the decision variables $x_{47,49}, x_{47,54}, x_{49,53}, x_{53,0}, x_{54,0}$. Therefore, we set $x_{47,49} = 1$ in the model and run the Simplex algorithm. We obtain an objective function $z = 133.55$ and the fractional variables $x_{0,14}, x_{0,15}, x_{14,16}, x_{15,22}, x_{16,22}$. We set the value $x_{0,14} = 1$ in the model and run the Simplex algorithm again. The objective function obtained is $z = 136.78$ and the fractional variables are $x_{14,16}, x_{14,23}, x_{16,22}, x_{22,29}, x_{23,29}$. We fix the decision variable $x_{14,16} = 1$ and executed the Simplex algorithm using this new restriction. We obtain a solution where all the decision variables take integer values, the objective function obtained is $z = 138.07$. The solution of the problem is $M = 10$ and ISE $= 8.07$. This solution is stored as the best feasible optimal found and is highlighted in Fig. 3. This figure presents a small piece of the tree generated by the Branch & Bound algorithm.

The algorithm should select a node to backtrack in order to continue. We select the last node which has been branched, that is, $x_{14,16}$. The algorithm selects now a different value $x_{14,16} = 0$ and solves the model using this new restriction. The objective function obtained is $z = 137.80$ and the decision variables with no integer values are $x_{14,17}, x_{14,23}, x_{17,22}, x_{22,29}, x_{23,29}$. The new value $x_{14,17} = 1$ is added to the model and solved. We obtain a solution where all decision variables are integer and an objective function $z = 139.20$, which is worse than the previous feasible solution found ($z = 138.07$). Therefore, this solution is not stored.

The algorithm selects the last node that has been branched to backtrack, and fixes a different possible value $x_{14,17} = 0$. The model is solved taking into account this new restriction. The solution contains fractional decision variables, however, the objective function $z = 139.59$ is worse than the best feasible solution found $z = 138.07$; therefore, no children of this node could improve the

360 best solution found and the node should be pruned.

361



362

Figure 3: This figure shows a little example (incomplete) of search tree generated by the Branch & Bound algorithm on the proposed model. The algorithm has been executed using
363 the contour chromosome and an error threshold $\varepsilon = 12$.

364 The algorithm keeps working as explained above, discarding those integer

solutions which have a worse objective function and pruning those nodes that have fractional variables, but whose objective function is worse than the best feasible solution found so far. The algorithm stops when no new nodes can be explored and the best feasible solution found is returned as the optimal solution.

4. Experiments and results

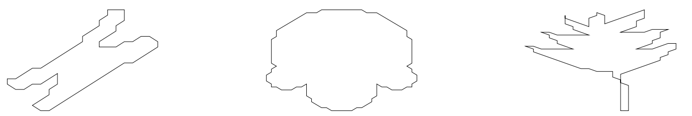
This section describes the experiments carried out to demonstrate the advantages of using the present proposal to solve the min-# polygonal approximation problem on closed curves. In the experiments we have used several state-of-the-art methods [19, 43, 27, 15, 21, 26, 7] to compare with out proposal.

For the experimentation we have used a generic computer with processor Intel(R) Core(TM) i7-3930K CPU @ 3.20GHz, 16 GB of RAM memory. The present proposal has been developed using C++ as the programming language and using Gurobi [44] (version 5.6.3) as the LP/MIP solver library. The default configuration for the MIP solver library was used. The modified version of the method proposed by Perez and Vidal [19] has been developed using C++ and taking into account that the distortion error can be computed in constant time $O(1)$ [19]. The method by Salotti [27] has been supplied by the author. The results of the other methods [43, 15, 21, 26, 7] have been obtained from the original papers.

4.1. Comparing optimal and suboptimal polygonal approximations

This experiment has been carried out to determine the optimality of the solution provided. We have used three synthetic contours commonly used in the literature [26, 21, 15, 43, 7] for comparing polygonal approximation algorithms. These synthetic contours are shown in Fig. 4.

Many of the algorithms that are defined in the literature obtain optimal solutions in several cases, however, the optimality of the solution is not assured. In this experiment we compare the results obtained by algorithms using genetic algorithms (GA [43], SMCR [21]), ant colony search algorithm (ACS [15]), graph



(a) Chromosome (N = 60) (b) Semicircle (N = 102) (c) Leaf (N = 120)

Figure 4: Synthetic contours

search approach (Betweenness [26]) and integer Particle Swarm Optimization (iPSO [7]).

We have also compared with the modified method proposed by Perez and Vidal [19]. This algorithm fixes the initial point of the curve to obtain the optimal polygonal approximation, however, the initial point of the curve may not be part of the optimal solution. Therefore, we must try all points of the curve as initial point in order to assure the optimality of the solution. We refer to the original way of using the algorithm as DP_0 (Dynamic Programming) and the approach which tries all points as initial point as DP_N. We have also used the optimal algorithm proposed by Salotti [27] in this experiment. This algorithm also fixes the initial point of the curve, therefore, we must try all points of the curve to assure the optimality of the solution. We refer to the original algorithm as $A_0^\star$ and the alternative that tries all points of the curve as initial point as $A_N^\star$. The results of our proposal (MIP) are also summarized in Table 1.

Table 1: This table shows the results for methods by Perez and Vidal [19] (using the first point of the curve as initial point DP_0, and trying all points as initial point DP_N), Salotti [27] (using the first point of the curve as initial node $A_0^\star$, and trying all points $A_N^\star$), Yin [43] (GA), Wang et al. [21] (SMCR), Yin [15] (ACS), Backes and Bruno [26] (Betweenness), Wang et al. [7] (iPSO) and our proposal (MIP). These results show that the only alternatives that obtain the optimal solution in every situation are DP_N, $A_N^\star$ and our proposal.

Contour	Method	ε	M	ISE	Merit
	Betweenness [26]	10	10	8.1	99.9

Chromosome

	ACS [15]	10	10	9.5	87.2
	GA [43]	10	11	9.15	82.6
	SMCR [21]	10	10	8.07	100
	DP_0 [19]	10	10	8.07	100
	DP_N [19]	10	10	8.07	100
	$A_0^\star$ [27]	10	10	8.07	100
	$A_N^\star$ [27]	10	10	8.07	100
	MIP	**10**	**10**	**8.07**	**100**
Semicircle	iPSO [7]	10	20	9.20	98.5
	SMCR [21]	10	20	9.68	94.9
	DP_0 [19]	10	20	9.01	100
	DP_N [19]	10	20	9.01	100
	$A_0^\star$ [27]	10	20	9.01	100
	$A_N^\star$ [27]	10	20	9.01	100
	MIP	**10**	**20**	**9.01**	**100**
	Betweenness [26]	20	14	19.8	91.2
	ACS [15]	20	16.4	19.9	72.6
	GA [43]	20	17	19.78	69.5
	SMCR [21]	20	14	18.16	97.0
	DP_0 [19]	20	14	17.39	100
	DP_N [19]	20	14	17.39	100
	$A_0^\star$ [27]	20	14	17.39	100
	$A_N^\star$ [27]	20	14	17.39	100
	MIP	**20**	**14**	**17.39**	**100**
Leaf	iPSO [7]	150	9	135.47	98.6
	Betweenness [26]	150	9	136.2	98.4
	ACS [15]	150	11.2	149.5	55.6
	GA [43]	150	12	149.46	52.0
	SMCR [21]	150	9	140.19	95.5
	DP_0 [19]	150	9	147.26	85.33

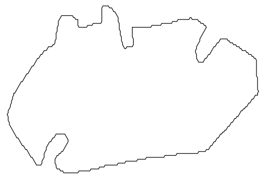
	DP_N [19]	150	8	141.58	100
	$A_0^\star$ [27]	150	9	147.26	85.33
	$A_N^\star$ [27]	150	8	141.58	100
	MIP	**150**	**8**	**141.58**	**100**

4.2. Comparing the methods for solving the min-# problem optimally

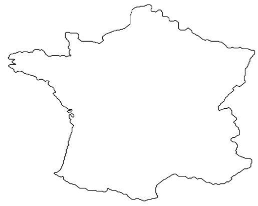
The proposed method solves the min-# problem optimally, hence, we compare our proposal with the methods that can assure the optimality of the solution. In the previous experiment, the methods proposed by Perez and Vidal [19] (DP_N), the alternative by Salotti [27] ($A_N^\star$) and our proposal, are the only methods which have obtained optimal solutions in all cases. We have also include the method proposed by Kolesnikov and Fränti [29] in this comparison although it can not be considered an optimal algorithm because the optimality of the solutions provided are not assured.

These alternatives obtain optimal solutions, therefore, it is not possible to demonstrate the advantages of our proposal by comparing the quality of the solutions. Due to the fact that the MIP optimization framework is Non-deterministic Polynomial-time (NP) we cannot supply the computational complexity for our method. Therefore, to assess the different proposals which obtain optimal solutions, we compare the computation time needed to obtain the optimal solution as was proposed in [27].

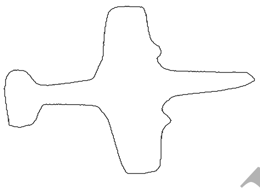
A set of real contours appearing in Fig. 5 has been used in this experiment (contour france and contour #1 are provided by Salotti [27]). These contours have been used to test the performance of the modified method by Perez and Vidal [19], the method based on graph search proposed by Salotti [27], the method by Kolesnikov and Fränti [29] and our proposal. We have used several distortion thresholds ε to obtain polygonal approximations using the different approaches. The results obtained are shown in Table 2.



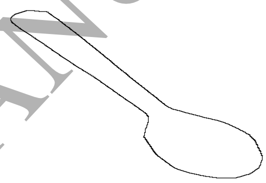
(a) Tinopener ($N = 581$)



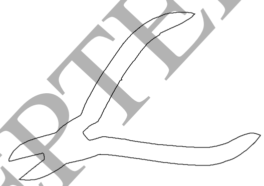
(b) France ($N = 992$)



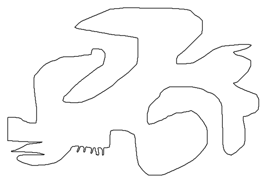
(c) Plane ($N = 1015$)



(d) Spoon ($N = 1370$)



(e) Pliers ($N = 2041$)

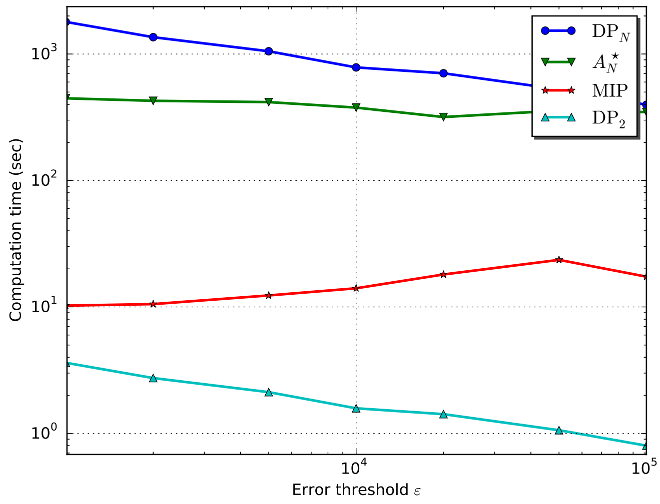


(f) Contour #1 ($N = 3222$)

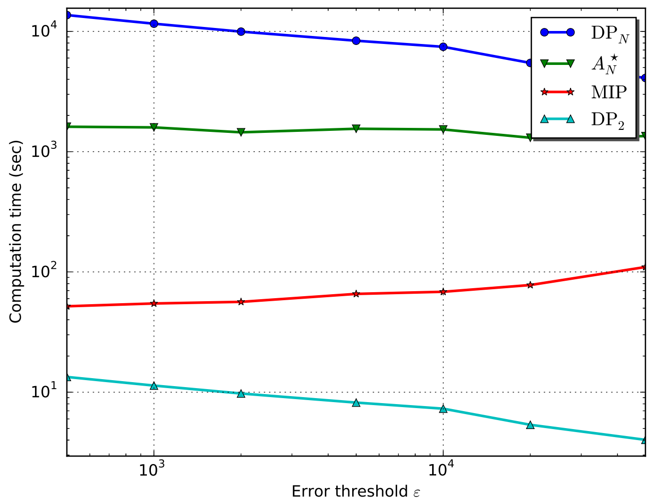
Figure 5: Contours dataset

Table 2: This table shows the computation times in seconds obtained using the method based on Dynamic Programming [19] (DP_N), the alternative proposed by Salotti [27] ($A^{\star}_N$), the algorithm proposed by Kolesnikov and Fränti [29] (DP_2) and our proposal (MIP); for solving the min-# problem on the set of contours which appear in Fig. 5 and using different values of the distortion threshold ε.

		ε						
Contour	Method	50	100	200	500	1000	2000	5000
Tinopener	DP_N [19]	794.6	632.2	533.6	429.2	417.6	359.6	336.4
	$A^{\star}_N$ [27]	348	371.1	330.6	301.6	255.2	336.4	307.4
	DP_2 [29]	2.74	2.18	1.84	1.48	1.44	1.24	1.16
	MIP	2.97	3.07	3.13	3.29	3.47	3.84	4.85
		1000	2000	5000	10000	20000	50000	100000
France	DP_N [19]	1795.5	1359	1051.5	783.7	704.3	525.7	396.8
	$A^{\star}_N$ [27]	446.4	426.6	416.6	376.9	317.4	357.1	347.2
	DP_2 [29]	3.62	2.74	2.12	1.58	1.41	1.05	0.8
	MIP	10.2	10.5	12.3	14	18.1	23.6	17.4
		200	500	1000	2000	5000	10000	20000
Plane	DP_N [19]	1755.9	1461.6	1187.5	1055.6	872.9	690.2	629.3
	$A^{\star}_N$ [27]	456.75	436.45	385.45	284.2	152.2	253.7	213.1
	DP_2 [29]	3.45	2.88	2.33	2.08	1.72	1.36	1.24
	MIP	10.1	10.1	10.5	11.3	12.4	12.6	13.8
		500	1000	2000	5000	10000	20000	50000
Spoon	DP_N [19]	2192	1863.2	1589.2	1452.2	1205.6	1109.7	1082.3
	$A^{\star}_N$ [27]	726.1	616.5	520.6	630.2	575.4	712.4	671.3
	DP_2 [29]	3.2	2.72	2.32	2.12	1.76	1.62	1.58
	MIP	28.7	28.6	33.3	35.7	53	56.1	41.2
		500	1000	2000	5000	10000	20000	50000
Pliers	DP_N [19]	13668	11587.2	9955.2	8364	7446	5467.2	4100.4
	$A^{\star}_N$ [27]	1611.6	1591.2	1448.4	1550.4	1530	1305.6	1346.4
	DP_2 [29]	13.39	11.35	9.76	8.20	7.30	5.36	4.02
	MIP	51.8	54.7	56.4	65.7	68.4	77.8	109.9
		20000	50000	100000	200000	500000	1000000	2000000
Contour #1	DP_N [19]	55933.9	45043.6	38470.7	31350.1	22779.5	18526.5	14144.6
	$A^{\star}_N$ [27]	10439.3	5122.9	5251.9	5445.2	5187.4	6057.4	6186.2
	DP_2 [29]	34.72	27.96	23.88	19.46	14.14	11.50	8.78
	MIP	171.3	180.2	210.8	336.4	186	237	263.1



(a) France



(b) Pliers

Figure 6: This figure shows the evolution of the computation time for the compared approaches (DP_N, $A_N^\star$, DP_2 and MIP) to solve the min-# problem optimally using different contours as input: (a) France, (b) Pliers. Both axes are plotted in logarithmic scale.

5. Discussion

In this section we discuss the most important aspects of the experiments carried out in this document. In Section 4.1 several state-of-the-art methods to solve the min-# problem have been compared with our proposal. That experiment shows that the state-of-the-art methods for solving the min-# problem obtain optimal results in some cases. For instance, the method by Wang et al. [21] obtains the optimal solution for contour chromosome using a value of $\varepsilon = 10$. However, none of the state-of-the-art methods [43, 15, 21, 26, 7] can assure the optimality of the solution. Nevertheless, the methods by Perez and Vidal [19], Salotti [27] and our proposal obtain the optimal solution in all cases.

Due to the fact that the optimality of the solutions is only assured for the method based on Dynamic Programming (DP_N), the method based on graph search ($A^\star_N$) and our proposal (MIP); we have performed another experiment to compare the performance of the optimal algorithms. We have included the method by Kolesnikov and Fränti [29] in this experiment, although can not be considered an optimal method. Table 2 shows the computation times obtained. These results show that our proposal obtains the optimal solution taking less computation time than the other optimal methods. The method DP_2 obtains lower computation times in all cases, however the optimality of the solution is not assured. The gap between the computation times of the method DP_2 and our proposal are due to the use of optimization methods (MIP) instead of a heuristic algorithm. The differences between the proposed method and the other optimal methods are remarkable as is shown in Fig. 6. This figure also shows that the differences between our proposal based on MIP and the method based on Dynamic Programming (DP_N) increase when decreasing the distortion threshold ε. For example, to solve the min-# problem for contour #1 using the distortion value $\varepsilon = 20000$ our proposal has proven to be more than 300 times faster than the method DP_N and more than 60 times faster than the method $A^\star_N$.

These results have shown that the computation time of our proposal are not so dependent on the threshold error ε. For example, the computation times appearing in Table 2 for the method based on Dynamic Programming and contour Plane have a standard deviation of $\pm 35.66\%$ regarding the mean value. However, the computation times obtained by the proposed method have a standard deviation of $\pm 9.73\%$ regarding the mean value, that is, do not change significantly on changing the value of the error threshold. This fact is also present in the method proposed by Salotti [27].

The main operations to compute the optimal solutions in the methods DP_N and $A^\star_N$ are the computation of the distortion (ISE) which can be done in constant time [19]. On the other hand, the Branch & Bound method uses the Simplex algorithm, which has polynomial-time average-case complexity [45, 46], to solve MIP models. Therefore, the differences between the computation time needed to obtain the optimal solutions by the different methods depends on the search space explored. Table 3 contains the search space explored to obtain the optimal solution. The method based on Dynamic Programming needs to check the complete search space to obtain optimal solutions. However the method $A^\star_N$ needs to check fewer nodes of the search space. Notice that the proposed alternative is the method which uses the fewest nodes to obtain the optimal polygonal approximation, and therefore needs to explore a small piece of the search space. Table 3 contains some values of 0 nodes explored for our proposal. This fact is due to the MIP model has been solved in the first stage of the process and all the integrality constraints are satisfied, and therefore, no nodes are explored by the Branch & Bound method. Notice that the first stage of the MIP solving is more time consuming than one Simplex iteration because some preprocessing is done (preprocessing of the decision variables, problem scaling, etc).

Table 3: This table shows the search space explored by the method based on Dynamic Programming [19] (DP_N), the alternative proposed by Salotti [27] ($A^\star_N$) and our proposal (MIP); for solving the min-# problem on the set of contours which appear in Fig. 5 and using different values of the distortion threshold ε. For the methods $A^\star_N$ and our proposal we have considered the nodes on the graph or the tree explored to obtain the optimal solution. The table shows that the search space explored by our proposal is smaller than the search space needed to explore by the other optimal methods.

		ε						
Contour	Method	50	100	200	500	1000	2000	5000
Tinopener	DP_N [19]	1.6×10^7	1.2×10^7	9.1×10^6	7.4×10^6	6.1×10^6	5.0×10^6	4.0×10^6
	$A^\star_N$ [27]	8.0×10^6	5.5×10^6	4.0×10^6	3.1×10^6	2.7×10^6	2.3×10^6	2.1×10^6
	MIP	0	0	0	0	1032	1291	0
		1000	2000	5000	10000	20000	50000	100000
France	DP_N [19]	4.6×10^7	3.2×10^7	2.1×10^7	1.5×10^7	1.2×10^7	7.9×10^6	5.9×10^6
	$A^\star_N$ [27]	2.2×10^7	1.5×10^7	9.2×10^6	6.3×10^6	5.7×10^6	3.5×10^6	2.6×10^6
	MIP	1220	0	0	0	3117	2931	2116
		200	500	1000	2000	5000	10000	20000
Plane	DP_N [19]	4.4×10^7	3.5×10^7	3.0×10^7	2.5×10^7	1.8×10^7	1.4×10^7	1.3×10^7
	$A^\star_N$ [27]	1.6×10^7	1.2×10^7	1.0×10^7	8.8×10^6	5.8×10^6	3.9×10^6	4.1×10^6
	MIP	0	0	0	1280	0	0	0
		500	1000	2000	5000	10000	20000	50000
Spoon	DP_N [19]	3.8×10^7	3.0×10^7	2.6×10^7	2.3×10^7	1.9×10^7	1.7×10^7	1.3×10^7
	$A^\star_N$ [27]	1.5×10^7	1.1×10^7	1.0×10^7	8.9×10^6	7.1×10^6	6.9×10^6	6.0×10^6
	MIP	1115	0	1228	0	329	1497	0
		500	1000	2000	5000	10000	20000	50000
Pliers	DP_N [19]	2.0×10^8	1.7×10^8	1.4×10^8	1.2×10^8	1.0×10^8	7.5×10^7	5.4×10^7
	$A^\star_N$ [27]	7.7×10^7	6.3×10^7	5.7×10^7	4.6×10^7	4.4×10^7	3.2×10^7	2.0×10^7
	MIP	0	2315	0	2159	1910	3233	0
		20000	50000	100000	200000	500000	1000000	2000000
Contour #1	DP_N [19]	5.3×10^8	4.3×10^8	3.6×10^8	2.9×10^8	2.1×10^8	1.7×10^8	1.2×10^8
	$A^\star_N$ [27]	2.1×10^8	1.7×10^8	1.6×10^8	1.3×10^8	9.3×10^7	8.1×10^7	6.5×10^7
	MIP	5129	0	2369	5688	0	2772	0

6. Conclusions

The current paper presents a novel and efficient method to obtain the minimum number of line segments for the polygonal approximation of a digital planar curve, using the ISE error criterion. It is based on the Mixed Integer

Programming optimization framework. The main idea is to represent all possible line segments and solutions using binary decision variables. Then, using a set of linear constraints, the MIP solver searches for the feasible optimal solution.

The present proposal has demonstrated to be faster than all optimal methods tested. Due to this issue, we recommend to use the present approach on complex and big contours, where the optimal polygonal approximation is required, for instance to compute measures to assess the quality of suboptimal polygonal approximations.

The present approach is capable to obtain the optimal solution on closed curves using the sum of square deviation criterion error. Our approach only needs one execution to assure the optimality of the solution in closed curves, because our proposal does not need to fix any point of the final solution as the initial point.

We have used the square deviation error to solve the min-# problem on closed curves. However our proposal could be easily adapted to use another error criterion.

Acknowledgments

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

We would like to thank Jean-Marc Salotti for providing the software and contours used in the experiments of this paper.

References

[1] Fred Attneave. Some informational aspects of visual perception. *Psychol. Rev*, pages 183–193, 1954.

[2] C.-H. Teh and R.T. Chin. On the detection of dominant points on digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):859–872, 1989. ISSN 0162-8828. doi: 10.1109/34.31447.

[3] Philippe Cornic. Another look at the dominant point detection of digital curves. *Pattern Recognition Letters*, 18(1):13–25, January 1997. ISSN 0167-8655. doi: 10.1016/S0167-8655(96)00116-X.

[4] Wen-Yen Wu. A dynamic method for dominant point detection. *Graphical Models*, 64(5):304–315, September 2002. ISSN 1524-0703. doi: 10.1016/S1077-3169(02)00008-4.

[5] Yi Xiao, Ju Jia Zou, and Hong Yan. An adaptive split-and-merge method for binary image contour data compression. *Pattern Recognition Letters*, 22(34):299–307, March 2001. ISSN 0167-8655. doi: 10.1016/S0167-8655(00)00138-0.

[6] Marc Salotti. An efficient algorithm for the optimal polygonal approximation of digitized curves. *Pattern Recognition Letters*, 22(2):215–221, February 2001. ISSN 0167-8655. doi: 10.1016/S0167-8655(00)00088-X.

[7] Bin Wang, Douglas Brown, Xiaozheng Zhang, Hanxi Li, Yongsheng Gao, and Jie Cao. Polygonal approximation using integer particle swarm optimization. *Information Sciences*, 278:311–326, September 2014. ISSN 0020-0255. doi: 10.1016/j.ins.2014.03.055.

[8] D. Dori and Wenyin Liu. Sparse pixel vectorization: an algorithm and its performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):202–215, March 1999. ISSN 0162-8828. doi: 10.1109/34.754586.

[9] David G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, pages 355–395, 1987.

[10] K. Grauman and T. Darrell. Fast contour matching using approximate earth mover's distance. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*, volume 1, pages I–220–I–227 Vol.1, June 2004. doi: 10.1109/CVPR.2004.1315035.

[11] Ömer M. Soysal and Jianhua Chen. Object recognition by spectral feature derived from canonical shape representation. *Machine Vision and Applications*, 24(4):855–868, December 2012. ISSN 0932-8092, 1432-1769. doi: 10.1007/s00138-012-0468-7.

[12] Minjie Chen, Mantao Xu, and P. Franti. A Fast Multiresolution Polygonal Approximation Algorithm for GPS Trajectory Simplification. *IEEE Transactions on Image Processing*, 21(5):2770–2785, May 2012. ISSN 1057-7149. doi: 10.1109/TIP.2012.2186146.

[13] David H. Douglas and Thomas 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):112–122, December 1973. doi: 10.3138/FM57-6770-U75U-7727.

[14] Peng-Yeng Yin. A new method for polygonal approximation using genetic algorithms. *Pattern Recognition Letters*, 19(11):1017–1026, September 1998. ISSN 0167-8655. doi: 10.1016/S0167-8655(98)00082-8.

[15] Peng-Yeng Yin. Ant colony search algorithms for optimal polygonal approximation of plane curves. *Pattern Recognition*, 36(8):1783–1797, August 2003. ISSN 0031-3203. doi: 10.1016/S0031-3203(02)00321-7.

[16] Mohammad Tanvir Parvez and Sabri A. Mahmoud. Polygonal approximation of digital planar curves through adaptive optimizations. *Pattern Recognition Letters*, 31(13):1997–2005, October 2010. ISSN 0167-8655. doi: 10.1016/j.patrec.2010.06.007.

[17] Asif Masood. Optimized polygonal approximation by dominant point deletion. *Pattern Recognition*, 41(1):227–239, January 2008. ISSN 0031-3203. doi: 10.1016/j.patcog.2007.05.021.

[18] Asif Masood. Dominant point detection by reverse polygonization of digital

curves. *Image and Vision Computing*, 26(5):702–715, May 2008. ISSN 0262-8856. doi: 10.1016/j.imavis.2007.08.006.

[19] Juan-Carlos Perez and Enrique Vidal. Optimum polygonal approximation of digitized curves. *Pattern Recognition Letters*, 15(8):743–750, August 1994. ISSN 0167-8655. doi: 10.1016/0167-8655(94)90002-7.

[20] E. J. Aguilera-Aguilera, A. Carmona-Poyato, F. J. Madrid-Cuevas, and R. Muñoz Salinas. Novel method to obtain the optimal polygonal approximation of digital planar curves based on Mixed Integer Programming. *Journal of Visual Communication and Image Representation*, 30:106–116, July 2015. ISSN 1047-3203. doi: 10.1016/j.jvcir.2015.03.007.

[21] Bin Wang, Huazhong Shu, and Limin Luo. A genetic algorithm with chromosome-repairing for min# and min polygonal approximation of digital curves. *Journal of Visual Communication and Image Representation*, 20 (1):45–56, January 2009. ISSN 1047-3203. doi: 10.1016/j.jvcir.2008.10.001.

[22] Jiahai Wang, Zhanghui Kuang, Xinshun Xu, and Yalan Zhou. Discrete particle swarm optimization based on estimation of distribution for polygonal approximation problems. *Expert Systems with Applications*, 36(5): 9398–9408, July 2009. ISSN 0957-4174. doi: 10.1016/j.eswa.2008.12.045.

[23] Peng-Yeng Yin. A discrete particle swarm algorithm for optimal polygonal approximation of digital curves. *Journal of Visual Communication and Image Representation*, 15(2):241–260, June 2004. ISSN 1047-3203. doi: 10.1016/j.jvcir.2003.12.001.

[24] Peng-Yeng Yin. Polygonal Approximation of Digital Curves Using the State-of-the-art Metaheuristics. In Goro Obinata and Ashish Dutt, editors, *Vision Systems: Segmentation and Pattern Recognition*. I-Tech Education and Publishing, June 2007. ISBN 978-3-902613-05-9.

[25] Arie Pikaz and Its'hak Dinstein. An algorithm for polygonal approximation

based on iterative point elimination. *Pattern Recognition Letters*, 16(6): 557–563, June 1995. ISSN 0167-8655. doi: 10.1016/0167-8655(95)80001-A.

[26] André Ricardo Backes and Odemir Martinez Bruno. Polygonal approximation of digital planar curves through vertex betweenness. *Information Sciences*, 222:795–804, February 2013. ISSN 0020-0255. doi: 10.1016/j.ins.2012.07.062.

[27] Marc Salotti. Optimal polygonal approximation of digitized curves using the sum of square deviations criterion. *Pattern Recognition*, 35(2):435–443, February 2002. ISSN 0031-3203. doi: 10.1016/S0031-3203(01)00051-6.

[28] Ji-Hwei Horng and Johnny T. Li. An automatic and efficient dynamic programming algorithm for polygonal approximation of digital curves. *Pattern Recognition Letters*, 23(13):171–182, January 2002. ISSN 0167-8655. doi: 10.1016/S0167-8655(01)00098-8.

[29] Alexander Kolesnikov and Pasi Fränti. Polygonal approximation of closed discrete curves. *Pattern Recognition*, 40(4):1282–1293, April 2007. ISSN 0031-3203. doi: 10.1016/j.patcog.2006.09.002.

[30] Debranjan Sarkar. A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves. *Pattern Recognition Letters*, 14(12):959–964, December 1993. ISSN 0167-8655. doi: 10.1016/0167-8655(93)90004-W.

[31] P.L. Rosin. Techniques for assessing polygonal approximations of curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6): 659–666, 1997. ISSN 0162-8828. doi: 10.1109/34.601253.

[32] A. Carmona-Poyato, R. Medina-Carnicer, F. J. Madrid-Cuevas, R. Muñoz Salinas, and N. L. Fernández-García. A new measurement for assessing polygonal approximation of curves. *Pattern Recognition*, 44(1):45–54, January 2011. ISSN 0031-3203. doi: 10.1016/j.patcog.2010.07.029.

[33] Augusto C. Rueda-Medina, John F. Franco, Marcos J. Rider, Antonio Padilha-Feltrin, and Rubén Romero. A mixed-integer linear programming approach for optimal type, size and allocation of distributed generation in radial distribution systems. *Electric Power Systems Research*, 97:133–143, April 2013. ISSN 0378-7796. doi: 10.1016/j.epsr.2012.12.009.

[34] Ying-Ming Wang and Peng Jiang. Alternative mixed integer linear programming models for identifying the most efficient decision making unit in data envelopment analysis. *Computers & Industrial Engineering*, 62(2): 546–553, March 2012. ISSN 0360-8352. doi: 10.1016/j.cie.2011.11.003.

[35] Adrián M. Aguirre, Carlos A. Méndez, Gloria Gutierrez, and Cesar De Prada. An improvement-based MILP optimization approach to complex AWS scheduling. *Computers & Chemical Engineering*, 47:217–226, December 2012. ISSN 0098-1354. doi: 10.1016/j.compchemeng.2012.06.036.

[36] S. Garrido-Jurado, R. Muñoz Salinas, F. J Madrid-Cuevas, and R. Medina-Carnicer. Generation of fiducial marker dictionaries using mixed integer linear programming. *Pattern Recognition*, 2015. ISSN 0031-3203. doi: 10.1016/j.patcog.2015.09.023.

[37] George Bernard Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963. ISBN 9780691080000.

[38] A. H. Land and A. G. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.

[39] Norman J. Driebeek. An Algorithm for the Solution of Mixed Integer Programming Problems. *Management Science*, 12(7):576–587, January 1966. ISSN 0025-1909, 1526-5501. doi: 10.1287/mnsc.12.7.576.

[40] J. A. Tomlin. Technical Note–An Improved Branch-and-Bound Method for Integer Programming. *Operations Research*, 19(4):1070–1075, January 1971. ISSN 0030-364X, 1526-5463. doi: 10.1287/opre.19.4.1070.

[41] J. Alemany, F. Magnago, D. Moitre, and H. Pinto. Symmetry issues in mixed integer programming based Unit Commitment. *International Journal of Electrical Power & Energy Systems*, 54:86–90, January 2014. ISSN 0142-0615. doi: 10.1016/j.ijepes.2013.06.034.

[42] Daniel T. Wojtaszek and John W. Chinneck. Faster MIP solutions via new node selection rules. *Computers & Operations Research*, 37(9):1544–1556, September 2010. ISSN 0305-0548. doi: 10.1016/j.cor.2009.11.011.

[43] Peng-Yeng Yin. Genetic algorithms for polygonal approximation of digital curves. *International Journal of Pattern Recognition and Artificial Intelligence*, 13(07):1061–1082, November 1999. ISSN 0218-0014. doi: 10.1142/S0218001499000598.

[44] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2015. URL `http://www.gurobi.com`.

[45] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, July 1998. ISBN 978-0-471-98232-6.

[46] Karl Heinz Borgwardt. *The Simplex Method*, volume 1 of *Algorithms and Combinatorics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1987. ISBN 978-3-540-17096-9 978-3-642-61578-8.

Chapter 6

Conclusions

Polygonal approximations are an active area of research in computer vision, because they are involved in many processes of the research area. These algorithms are a key phase on vectorization, due to polygonal approximation methods simplifies the polyline obtained in the previous phase. Polygonal approximation methods are also used to obtain feature vectors for shape matching and object recognition. These methods are also very popular for compressing shape and contour representations.

This thesis has proposed three main contributions in the area of polygonal approximations. The following objectives has been achieved:

- Some polygonal approximation methods use an initial set of points (e.g. the breakpoints) to obtain the polygonal approximation. In [1], we have introduce a new subset of initial points, named CDP (Candidate Dominant Points), that boost the polygonal approximation methods and also make these methods to improve the quality of the approximation in some cases.

- In [2], we have proposed an optimal method to solve the min-ε polygonal approximation problem using the L_2-norm as the distortion measure. The optimal polygonal approximation is used as a reference solution for using frameworks for assessing the quality of suboptimal polygonal approximations. However obtaining this optimal solution is a complex process and also the computation burden is very high. Our method, which is based on Mixed Integer Programming (MIP) obtains the optimal solution is all cases. The proposed method is faster than all the optimal methods tested.

- We have also solved optimally the min-ε polygonal approximation problem using a novel proposal. In [9], we have proposed a new algorithm with two stages. Firstly, the method proposed by Pikaz and Dinstein [38] is used to obtain a suboptimal polygonal approximation. Then, the improved Salotti's method is used to obtain several local optimal solutions with a prefixed starting point. In each iteration the distortion error is used as the threshold value to prune suboptimal solutions. Several strategies to select the starting points has been tested.

- Finally, in [3], an optimal algorithm to solve the min-# polygonal approximation problem is proposed. This method also uses the Mixed Integer Programming framework to obtain the optimal solution.

We want to remark other additional contributions obtained. For instance, in [4] we propose a method to determine an appropriate number of line segments that a polygonal approximation needs to represent faithfully any planar digital curve. Another polygonal approximation method was proposed in [10].

This thesis has focused on contributions of polygonal approximation methods. A future research area could be focused on the usage of polygonal approximations. In this document several usages for polygonal approximations have been pointed out: contour and shape recognition, shape analysis, computer vision auxiliary methods and object placement recognition.

Bibliography

[1] E. J. Aguilera-Aguilera, A. Carmona-Poyato, F. J. Madrid-Cuevas, and R. Medina-Carnicer. The computation of polygonal approximations for 2d contours based on a concavity tree. *Journal of Visual Communication and Image Representation*, 25(8): 1905–1917, November 2014. ISSN 1047-3203. doi: 10.1016/j.jvcir.2014.09.012. URL `http://dx.doi.org/10.1016/j.jvcir.2014.09.012`.

[2] E. J. Aguilera-Aguilera, A. Carmona-Poyato, F. J. Madrid-Cuevas, and R. Muñoz-Salinas. Novel method to obtain the optimal polygonal approximation of digital planar curves based on Mixed Integer Programming. *Journal of Visual Communication and Image Representation*, 30:106–116, July 2015. ISSN 1047-3203. doi: 10.1016/j.jvcir.2015.03.007. URL `http://dx.doi.org/10.1016/j.jvcir.2015.03.007`.

[3] E. J. Aguilera-Aguilera, A. Carmona-Poyato, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. Fast computation of optimal polygonal approximations of digital planar closed curves. *Graphical Models*, April 2016. ISSN 1524-0703. doi: 10.1016/j.gmod.2016.01.004. URL `http://dx.doi.org/10.1016/j.gmod.2016.01.004`.

[4] Eusebio J. Aguilera-Aguilera, Angel Carmona-Poyato, Francisco J. Madrid-Cuevas, and Rafael Medina-Carnicer. Unsupervised Approximation of Digital Planar Curves. In Roberto Paredes, Jaime S. Cardoso, and Xosé M. Pardo, editors, *Pattern Recognition and Image Analysis*, number 9117 in Lecture Notes in Computer Science, pages 200–207. Springer International Publishing, June 2015. ISBN 978-3-319-19389-2 978-3-319-19390-8. URL `http://dx.doi.org/10.1007/978-3-319-19390-8_23`.

[5] Nirwan Ansari and Edward J. Delp. On detecting dominant points. *Pattern Recognition*, 24(5):441–451, 1991. ISSN 0031-3203. doi: 10.1016/0031-3203(91)90057-C. URL `http://dx.doi.org/10.1016/0031-3203(91)90057-C`.

[6] A. Carmona-Poyato, N.L. Fernández-García, R. Medina-Carnicer, and F.J. Madrid-Cuevas. Dominant point detection: A new proposal. *Image and Vision Computing*, 23 (13):1226–1236, November 2005. ISSN 0262-8856. doi: 10.1016/j.imavis.2005.07.025. URL `http://dx.doi.org/10.1016/j.imavis.2005.07.025`.

[7] 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(1):14–25, January 2010. ISSN 0031-3203. doi: 10.1016/j.patcog.2009.06.010. URL `http://dx.doi.org/10.1016/j.patcog.2009.06.010`.

[8] A. Carmona-Poyato, R. Medina-Carnicer, F. J. Madrid-Cuevas, R. Muñoz-Salinas, and N. L. Fernández-García. A new measurement for assessing polygonal approximation of curves. *Pattern Recognition*, 44(1):45–54, January 2011. ISSN 0031-3203. doi: 10.1016/j.patcog.2010.07.029. URL `http://dx.doi.org/10.1016/j.patcog.2010.07.029`.

[9] A. Carmona-Poyato, E.J. Aguilera-Aguilera, F.J. Madrid-Cuevas, M.J. Marín-Jiménez, and N.L. Fernández-García. New method for obtaining optimal polygonal approximations to solve the min-ε problem. *Neural Computing and Applications*, pages 1–12, February 2016. doi: 10.1007/s00521-016-2198-7. URL `http://dx.doi.org/10.1007/s00521-016-2198-7`.

[10] Angel Carmona-Poyato, Eusebio J. Aguilera-Aguilera, Francisco J. Madrid-Cuevas, and D. López-Fernandez. New Method for Obtaining Optimal Polygonal Approximations. In Roberto Paredes, Jaime S. Cardoso, and Xosé M. Pardo, editors, *Pattern Recognition and Image Analysis*, number 9117 in Lecture Notes in Computer Science, pages 149–156. Springer International Publishing, June 2015. ISBN 978-3-319-19389-2 978-3-319-19390-8. URL `http://dx.doi.org/10.1007/978-3-319-19390-8_17`.

[11] Jen-Ming Chen, Jose A. Ventura, and Chih-Hang Wu. Segmentation of planar curves into circular arcs and line segments. *Image and Vision Computing*, 14(1):71–83, February 1996. ISSN 0262-8856. doi: 10.1016/0262-8856(95)01042-4. URL `http://dx.doi.org/10.1016/0262-8856(95)01042-4`.

[12] David H. Douglas and Thomas 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):112–122, December 1973. doi: 10.3138/FM57-6770-U75U-7727. URL `http://dx.doi.org/10.3138/FM57-6770-U75U-7727`.

[13] J.G. Dunham. Optimum Uniform Piecewise Linear Approximation of Planar Curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1):67–75, 1986. ISSN 0162-8828. doi: 10.1109/TPAMI.1986.4767753. URL `http://dx.doi.org/10.1109/TPAMI.1986.4767753`.

[14] Liu Guanghui and Chen Chuanbo. A New Approach for Polygonal Approximation of Shape Contours using Genetic Algorithm. In *2nd IEEE Conference on Industrial Electronics and Applications, 2007. ICIEA 2007*, pages 763–768, May 2007. doi: 10.1109/ICIEA.2007.4318510. URL `http://dx.doi.org/10.1109/ICIEA.2007.4318510`.

[15] Ji-Hwei Horng and Johnny T. Li. An automatic and efficient dynamic programming algorithm for polygonal approximation of digital curves. *Pattern Recognition Letters*, 23 (1–3):171–182, January 2002. ISSN 0167-8655. doi: 10.1016/S0167-8655(01)00098-8. URL `http://dx.doi.org/10.1016/S0167-8655(01)00098-8`.

[16] Shu-Chien Huang. Polygonal Approximation Using an Artificial Bee Colony Algorithm. *Mathematical Problems in Engineering*, 2015:10, February 2015. ISSN 1024-123X. doi: 10.1155/2015/375926. URL `http://dx.doi.org/10.1155/2015/375926`.

[17] Shu-Chien Huang and Yung-Nien Sun. Polygonal approximation using genetic algorithms. *Pattern Recognition*, 32(8):1409–1420, August 1999. ISSN 0031-3203. doi: 10.1016/S0031-3203(98)00173-3. URL `http://dx.doi.org/10.1016/S0031-3203(98)00173-3`.

[18] Hiroshi IMAI and Masao IRI. Polygonal Approximations of a Curve — Formulations and Algorithms. In Godfried T. TOUSSAINT, editor, *Machine Intelligence and Pattern Recognition*, volume Volume 6, pages 71–86. North-Holland, 1988. ISBN 0923-0459. URL `http://dx.doi.org/10.1016/B978-0-444-70467-2.50011-4`.

[19] A. Kolesnikov. Nonparametric polygonal and multimodel approximation of digital curves with Rate-Distortion curve modeling. pages 2889–2892, 2011. doi: 10.1109/ICIP.2011.6116152. URL `http://dx.doi.org/10.1109/ICIP.2011.6116152`.

[20] Alexander Kolesnikov and Tuomo Kauranne. Unsupervised segmentation and approximation of digital curves with rate-distortion curve modeling. *Pattern Recognition*, 47 (2):623–633, February 2014. ISSN 0031-3203. doi: 10.1016/j.patcog.2013.09.002. URL `http://dx.doi.org/10.1016/j.patcog.2013.09.002`.

[21] F. Lecron, M. Benjelloun, and S. Mahmoudi. Points of interest detection in cervical spine radiographs by polygonal approximation. In *2010 2nd International Conference on Image Processing Theory Tools and Applications (IPTA)*, pages 81–86, July 2010. doi: 10.1109/IPTA.2010.5586771. URL `http://dx.doi.org/10.1109/IPTA.2010.5586771`.

[22] Herve Locteau, Romain Raveaux, Sebastien Adam, Yves Lecourtier, Pierre Heroux, and Eric Trupin. Polygonal Approximation of Digital Curves Using a Multi-objective Genetic Algorithm. In Wenyin Liu and Josep Lladós, editors, *Graphics Recognition. Ten Years Review and Future Perspectives*, number 3926 in Lecture Notes in Computer Science, pages 300–311. Springer Berlin Heidelberg, January 2006. ISBN 978-3-540-34711-8, 978-3-540-34712-5. URL `http://dx.doi.org/10.1007/11767978_27`.

[23] David G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, pages 355–395, 1987. URL `http://dx.doi.org/10.1016/0004-3702(87)90070-1`.

[24] Majed Marji and Pepe Siy. Polygonal representation of digital planar curves through dominant point detection—a nonparametric algorithm. *Pattern Recognition*, 37(11):2113–2130, November 2004. ISSN 0031-3203. doi: 10.1016/j.patcog.2004.03.004. URL http://dx.doi.org/10.1016/j.patcog.2004.03.004.

[25] Asif Masood. Dominant point detection by reverse polygonization of digital curves. *Image and Vision Computing*, 26(5):702–715, May 2008. ISSN 0262-8856. doi: 10.1016/j.imavis.2007.08.006. URL http://dx.doi.org/10.1016/j.imavis.2007.08.006.

[26] Asif Masood. Optimized polygonal approximation by dominant point deletion. *Pattern Recognition*, 41(1):227–239, January 2008. ISSN 0031-3203. doi: 10.1016/j.patcog.2007.05.021. URL http://dx.doi.org/10.1016/j.patcog.2007.05.021.

[27] Asif Masood and Shaiq A. Haq. A novel approach to polygonal approximation of digital curves. *Journal of Visual Communication and Image Representation*, 18(3):264–274, June 2007. ISSN 1047-3203. doi: 10.1016/j.jvcir.2006.12.002. URL http://dx.doi.org/10.1016/j.jvcir.2006.12.002.

[28] Avraham Melkman and Joseph O'Rourke. On Polygonal Chain Approximation. In Godfried T. TOUSSAINT, editor, *Machine Intelligence and Pattern Recognition*, volume Volume 6, pages 87–95. North-Holland, 1988. ISBN 0923-0459. doi: 10.1016/B978-0-444-70467-2.50012-6. URL http://dx.doi.org/10.1016/B978-0-444-70467-2.50012-6.

[29] D.A. Mitzias and B.G. Mertzios. Shape recognition with a neural classifier based on a fast polygon approximation technique. *Pattern Recognition*, 27(5):627–636, May 1994. ISSN 0031-3203. doi: 10.1016/0031-3203(94)90042-6. URL http://dx.doi.org/10.1016/0031-3203(94)90042-6.

[30] Yasuo Nakagawa and Azriel Rosenfeld. A note on polygonal and elliptical approximation of mechanical parts. *Pattern Recognition*, 11(2):133–142, 1979. ISSN 0031-3203. doi: 10.1016/0031-3203(79)90059-1. URL http://dx.doi.org/10.1016/0031-3203(79)90059-1.

[31] Thanh Phuong Nguyen and Isabelle Debled-Rennesson. Parameter-free method for polygonal representation of the noisy curves. In *International Workshop on Combinatorial Image Analysis*, pages 65–78, Mexico, November 2009. RPS. URL https://hal.archives-ouvertes.fr/hal-00437309.

[32] G. Papakonstantinou. Optimal polygonal approximation of digital curves. *Signal Processing*, 8(1):131–135, February 1985. ISSN 0165-1684. doi: 10.1016/0165-1684(85)90094-5. URL http://dx.doi.org/10.1016/0165-1684(85)90094-5.

[33] Mohammad Tanvir Parvez and Sabri A. Mahmoud. Polygonal approximation of digital planar curves through adaptive optimizations. *Pattern Recognition Letters*, 31(13):1997–

2005, October 2010. ISSN 0167-8655. doi: 10.1016/j.patrec.2010.06.007. URL http://dx.doi.org/10.1016/j.patrec.2010.06.007.

[34] Theodosios Pavlidis and S.L. Horowitz. Segmentation of Plane Curves. *IEEE Transactions on Computers*, C-23(8):860–870, 1974. ISSN 0018-9340. doi: 10.1109/T-C.1974.224041. URL http://dx.doi.org/10.1109/T-C.1974.224041.

[35] M. Pawan Kumar, Saurabh Goyal, Sujit Kuthirummal, C. V. Jawahar, and P. J. Narayanan. Discrete contours in multiple views: approximation and recognition. *Image and Vision Computing*, 22(14):1229–1239, December 2004. ISSN 0262-8856. doi: 10.1016/j.imavis.2004.03.022. URL http://dx.doi.org/10.1016/j.imavis.2004.03.022.

[36] Soo-Chang Pei and Ji-Hwei Horng. Optimum approximation of digital planar curves using circular arcs. *Pattern Recognition*, 29(3):383–388, March 1996. ISSN 0031-3203. doi: 10.1016/0031-3203(95)00104-2. URL http://dx.doi.org/10.1016/0031-3203(95)00104-2.

[37] Juan-Carlos Perez and Enrique Vidal. Optimum polygonal approximation of digitized curves. *Pattern Recognition Letters*, 15(8):743–750, August 1994. ISSN 0167-8655. doi: 10.1016/0167-8655(94)90002-7. URL http://dx.doi.org/10.1016/0167-8655(94)90002-7.

[38] Arie Pikaz and Its'hak Dinstein. An algorithm for polygonal approximation based on iterative point elimination. *Pattern Recognition Letters*, 16(6):557–563, June 1995. ISSN 0167-8655. doi: 10.1016/0167-8655(95)80001-A. URL http://dx.doi.org/10.1016/0167-8655(95)80001-A.

[39] Arie Pikaz and Its'hak Dinstein. Optimal polygonal approximation of digital curves. *Pattern Recognition*, 28(3):373–379, March 1995. ISSN 0031-3203. doi: 10.1016/0031-3203(94)00108-X. URL http://dx.doi.org/10.1016/0031-3203(94)00108-X.

[40] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256, November 1972. ISSN 0146-664X. doi: 10.1016/S0146-664X(72)80017-0. URL http://dx.doi.org/10.1016/S0146-664X(72)80017-0.

[41] Bimal Kr. Ray and Kumar S. Ray. A non-parametric sequential method for polygonal approximation of digital curves. *Pattern Recognition Letters*, 15(2):161–167, February 1994. ISSN 0167-8655. doi: 10.1016/0167-8655(94)90045-0. URL http://dx.doi.org/10.1016/0167-8655(94)90045-0.

[42] Paul L Rosin and Geoff AW West. Segmentation of edges into lines and arcs. *Image and Vision Computing*, 7(2):109–114, May 1989. ISSN 0262-8856. doi: 10.1016/0262-8856(89)90004-8. URL http://dx.doi.org/10.1016/0262-8856(89)90004-8.

[43] P.L. Rosin. Techniques for assessing polygonal approximations of curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):659–666, 1997. ISSN 0162-8828. doi: 10.1109/34.601253. URL http://dx.doi.org/10.1109/34.601253.

[44] Marc Salotti. An efficient algorithm for the optimal polygonal approximation of digitized curves. *Pattern Recognition Letters*, 22(2):215–221, February 2001. ISSN 0167-8655. doi: 10.1016/S0167-8655(00)00088-X. URL http://dx.doi.org/10.1016/S0167-8655(00)00088-X.

[45] Marc Salotti. Optimal polygonal approximation of digitized curves using the sum of square deviations criterion. *Pattern Recognition*, 35(2):435–443, February 2002. ISSN 0031-3203. doi: 10.1016/S0031-3203(01)00051-6. URL http://dx.doi.org/10.1016/S0031-3203(01)00051-6.

[46] Biswajit Sarkar, Lokendra K. Singh, and Debranjan Sarkar. Approximation of digital curves with line segments and circular arcs using genetic algorithms. *Pattern Recognition Letters*, 24(15):2585–2595, November 2003. ISSN 0167-8655. doi: 10.1016/S0167-8655(03)00103-X. URL http://dx.doi.org/10.1016/S0167-8655(03)00103-X.

[47] Debranjan Sarkar. A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves. *Pattern Recognition Letters*, 14(12):959–964, December 1993. ISSN 0167-8655. doi: 10.1016/0167-8655(93)90004-W. URL http://dx.doi.org/10.1016/0167-8655(93)90004-W.

[48] Yukio Sato. Piecewise linear approximation of plane curves by perimeter optimization. *Pattern Recognition*, 25(12):1535–1543, December 1992. ISSN 0031-3203. doi: 10.1016/0031-3203(92)90126-4. URL http://dx.doi.org/10.1016/0031-3203(92)90126-4.

[49] Jack Sklansky and Victor Gonzalez. Fast polygonal approximation of digitized curves. *Pattern Recognition*, 12(5):327–331, 1980. ISSN 0031-3203. doi: 10.1016/0031-3203(80)90031-X. URL http://dx.doi.org/10.1016/0031-3203(80)90031-X.

[50] Andrés Solís Montero and Jochen Lang. Skeleton pruning by contour approximation and the integer medial axis transform. *Computers & Graphics*, 36(5):477–487, August 2012. ISSN 0097-8493. doi: 10.1016/j.cag.2012.03.029. URL http://dx.doi.org/10.1016/j.cag.2012.03.029.

[51] Jan Stria, Daniel Průša, and Václav Hlaváč. Polygonal Models for Clothing. In Michael Mistry, Aleš Leonardis, Mark Witkowski, and Chris Melhuish, editors, *Advances in Autonomous Robotics Systems*, number 8717 in Lecture Notes in Computer Science, pages 173–184. Springer International Publishing, September 2014. ISBN 978-3-319-10400-3 978-3-319-10401-0. URL http://dx.doi.org/10.1007/978-3-319-10401-0_16.

[52] Mohammad Tanvir Parvez and Sabri A. Mahmoud. Arabic handwriting recognition using structural and syntactic pattern attributes. *Pattern Recognition*, 46(1):141–154, January 2013. ISSN 0031-3203. doi: 10.1016/j.patcog.2012.07.012. URL `http://dx.doi.org/10.1016/j.patcog.2012.07.012`.

[53] C.-H. Teh and R.T. Chin. On the detection of dominant points on digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):859–872, 1989. ISSN 0162-8828. doi: 10.1109/34.31447. URL `http://dx.doi.org/10.1109/34.31447`.

[54] Chien-Cheng Tseng, Chang-Jung Juan, Hsi-Cheng Chang, and Jeen-Fong Lin. An optimal line segment extraction algorithm for online Chinese character recognition using dynamic programming. *Pattern Recognition Letters*, 19(10):953–961, August 1998. ISSN 0167-8655. doi: 10.1016/S0167-8655(98)00071-3. URL `http://dx.doi.org/10.1016/S0167-8655(98)00071-3`.

[55] Karin Wall and Per-Erik Danielsson. A fast sequential method for polygonal approximation of digitized curves. *Computer Vision, Graphics, and Image Processing*, 28(2):220–227, November 1984. ISSN 0734-189X. doi: 10.1016/S0734-189X(84)80023-7. URL `http://dx.doi.org/10.1016/S0734-189X(84)80023-7`.

[56] Wenhua Wan and José A. Ventura. Segmentation of Planar Curves into Straight-Line Segments and Elliptical Arcs. *Graphical Models and Image Processing*, 59(6):484–494, November 1997. ISSN 1077-3169. doi: 10.1006/gmip.1997.0450. URL `http://dx.doi.org/10.1006/gmip.1997.0450`.

[57] Bin Wang, Hua-Zhong Shu, Bao-Sheng Li, and Zhi-Mei Niu. A Mutation-Particle Swarm Algorithm for Error-Bounded Polygonal Approximation of Digital Curves. In De-Shuang Huang, Donald C. Wunsch II, Daniel S. Levine, and Kang-Hyun Jo, editors, *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues*, pages 1149–1155. Springer Berlin Heidelberg, January 2008. ISBN 978-3-540-87440-9, 978-3-540-87442-3. URL `http://dx.doi.org/10.1007/978-3-540-87442-3_142`.

[58] Bin Wang, Huazhong Shu, Chaojian Shi, and Limin Luo. A novel stochastic search method for polygonal approximation problem. *Neurocomputing*, 71(16–18):3216–3223, October 2008. ISSN 0925-2312. doi: 10.1016/j.neucom.2008.04.028. URL `http://dx.doi.org/10.1016/j.neucom.2008.04.028`.

[59] Bin Wang, Huazhong Shu, and Limin Luo. A genetic algorithm with chromosome-repairing for min-# and min-ε polygonal approximation of digital curves. *Journal of Visual Communication and Image Representation*, 20(1):45–56, 2009. ISSN 1047-3203. doi: 10.1016/j.jvcir.2008.10.001. URL `http://dx.doi.org/10.1016/j.jvcir.2008.10.001`.

[60] Bin Wang, Douglas Brown, Xiaozheng Zhang, Hanxi Li, Yongsheng Gao, and Jie Cao. Polygonal approximation using integer particle swarm optimization. *Information Sciences*, 278:311–326, September 2014. ISSN 0020-0255. doi: 10.1016/j.ins.2014.03.055. URL `http://dx.doi.org/10.1016/j.ins.2014.03.055`.

[61] Jiahai Wang, Zhanghui Kuang, Xinshun Xu, and Yalan Zhou. Discrete particle swarm optimization based on estimation of distribution for polygonal approximation problems. *Expert Systems with Applications*, 36(5):9398–9408, July 2009. ISSN 0957-4174. doi: 10.1016/j.eswa.2008.12.045. URL `http://dx.doi.org/10.1016/j.eswa.2008.12.045`.

[62] Liu Wenyin and Dov Dori. From Raster to Vectors: Extracting Visual Information from Line Drawings. *Pattern Analysis & Applications*, 2(1):10–21, February 2014. ISSN 1433-7541. doi: 10.1007/s100440050010. URL `http://dx.doi.org/10.1007/s100440050010`.

[63] Yi Xiao, Ju Jia Zou, and Hong Yan. An adaptive split-and-merge method for binary image contour data compression. *Pattern Recognition Letters*, 22(3–4):299–307, March 2001. ISSN 0167-8655. doi: 10.1016/S0167-8655(00)00138-0. URL `http://dx.doi.org/10.1016/S0167-8655(00)00138-0`.

[64] Peng-Yeng Yin. A new method for polygonal approximation using genetic algorithms. *Pattern Recognition Letters*, 19(11):1017–1026, September 1998. ISSN 0167-8655. doi: 10.1016/S0167-8655(98)00082-8. URL `http://dx.doi.org/10.1016/S0167-8655(98)00082-8`.

[65] Peng-Yeng Yin. Ant colony search algorithms for optimal polygonal approximation of plane curves. *Pattern Recognition*, 36(8):1783–1797, August 2003. ISSN 0031-3203. doi: 10.1016/S0031-3203(02)00321-7. URL `http://dx.doi.org/10.1016/S0031-3203(02)00321-7`.

[66] Peng-Yeng Yin. A discrete particle swarm algorithm for optimal polygonal approximation of digital curves. *Journal of Visual Communication and Image Representation*, 15(2):241–260, June 2004. ISSN 1047-3203. doi: 10.1016/j.jvcir.2003.12.001. URL `http://dx.doi.org/10.1016/j.jvcir.2003.12.001`.

[67] Hongbin Zhang and Jianjun Guo. Optimal polygonal approximation of digital planar curves using meta heuristics. *Pattern Recognition*, 34(7):1429–1436, 2001. ISSN 0031-3203. doi: 10.1016/S0031-3203(00)00097-2. URL `http://dx.doi.org/10.1016/S0031-3203(00)00097-2`.

Impact report

This report shows the impact factors and positions of the journals in which the works presented in this thesis have been published.

- E. J. Aguilera-Aguilera, A. Carmona-Poyato, F. J. Madrid-Cuevas, and R. Medina-Carnicer. The computation of polygonal approximations for 2d contours based on a concavity tree. Journal of Visual Communication and Image Representation, pages 1905-1917, November 2014.
  Journal: Journal of Visual Communication and Image Representation
  Impact factor: JCR 2014: 1.218
  Position of the journal: 57 of 139 (Q2)
  Category: COMPUTER SCIENCE, INFORMATION SYSTEMS

- E. J. Aguilera-Aguilera, A. Carmona-Poyato, F. J. Madrid-Cuevas, and R. Muñoz-Salinas. Novel method to obtain the optimal polygonal approximation of digital planar curves based on Mixed Integer Programming. Journal of Visual Communication and Image Representation, pages 106-116, July 2015.
  Journal: Journal of Visual Communication and Image Representation
  Impact factor: JCR 2014: 1.218
  Position of the journal: 57 of 139 (Q2)
  Category: COMPUTER SCIENCE, INFORMATION SYSTEMS

- A. Carmona-Poyato, E. J. Aguilera-Aguilera, F. J. Madrid-Cuevas, M. J. Marín-Jiménez, and N. L. Fernández-García. New method for obtaining optimal polygonal approximations to solve the min-ε problem. Neural Computing and Applications, pages 1-12, February 2016.
  Journal: Neural Computing and Applications
  Impact factor: JCR 2014: 1.569
  Position of the journal: 53 of 123 (Q2)
  Category: COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE

- E. J. Aguilera-Aguilera, A. Carmona-Poyato, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. Fast computation of optimal polygonal approximations of digital planar closed curves. Graphical Models, April 2016.
  Journal: Graphical Models
  Impact Factor: JCR 2014: 1.049
  Position of the journal: 45 of 104 (Q2)
  Category: COMPUTER SCIENCE, SOFTWARE ENGINEERING