Acta Universitaria
ISSN: 0188-6266
actauniversitaria@ugto.mx
Universidad de Guanajuato
México

Alvarado-Velazco, Paola B.; Ayala-Ramirez, Victor; Sanchez-Yanez, Raul E.

Polygonal Approximation of Digital Curves Using Evolutionary Programming

Acta Universitaria, vol. 22, marzo, 2012, pp. 15-20
Universidad de Guanajuato
Guanajuato, México

Acta Universitaria
DIRECCIÓN DE APOYO A LA INVESTIGACIÓN Y AL POSGRADO

Polygonal Approximation of Digital Curves Using Evolutionary Programming

Paola B. Alvarado-Velazco*, Victor Ayala-Ramirez* and Raul E. Sanchez-Yanez*

ABSTRACT

This paper proposes an Evolutionary Programming (EP) approach to solve the polygonal approximation of digital curves. The solution provided by the method consists of a sequence of straight line segments to be applied as Advance and Rotate motion primitives of a 2D Cartesian robot. The proposed approach finds automatically the number of segments and the starting and ending points of each of them. We have tested our approach on a test set of digital curves that exhibits two main qualitative features: openess and straightness, in different degrees. We show that our method obtains good results for approximating the curves in the test set. We present both quantitative and qualitative results of these test.

RESUMEN

El artículo propone un método basado en Programación Evolutiva para solucionar la aproximación poligonal de curvas digitales. La solución proporcionada por el método consiste en una secuencia de segmentos de línea que será traducida a primitivas de movimiento Avanzar y Rotar en un robot Cartesiano 2D. El método propuesto encuentra automáticamente el número de segmentos así como sus puntos iniciales y finales. El método ha sido probado en un conjunto de curvas digitales de prueba que exhibe principalmente dos características: openess y straightness, en diferentes grados. Se demuestra que el método propuesto proporciona buenos resultados para la aproximación de curvas en el conjunto de pruebas. Presentamos resultados cualitativos y cuantitativos de estas pruebas.

Recibido: 12 Enero 2012
Aceptado: 12 Febrero 2012

INTRODUCTION

Shape approximation has received a lot of research attention because of its multiple applications. Some examples of these applications are: its use to optimize machine elements, the simplification of CAD models and the approximation of trajectory data. Main approaches to solve the shape approximation problem are the approximation of shapes using polygonal segments and the use of splines for the same purpose. In particular, we are interested in polygonal approximation of digital curves because we intend to synthesize the trajectory of a 2D Cartesian robot using only *Advance* and *Rotate* primitives along an arbitrary 2D curve.

RELATED WORK

Polygonal approximation of digital curves has been addressed by many researchers using different methods that could be categorized into three main classes [1][2]: *i)* local optimization algorithms , *ii)* global optimization algorithms and *iii)* intelligent optimization algorithms.

Local optimization algorithms include approaches using sequential methods, split and merge techniques and dominant point detection methods. Global optimization techniques use dynamic programming methods. Both categories of methods present several drawbacks in the general case. Intelligent optimization algorithms, such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO), has been gained attention as alternatives to solve the polygonal approximation problem because they can search efficiently through complex search spaces such as the ones generated by the approximation of complex digital curves.

Palabras clave:
Aproximación poligonal; curvas digitales; programación evolutiva; optimización.
Keywords:
Polygonal approximation; digital curves; evolutionary programming, optimization.

---

* Electronics Engineering Department, Universidad de Guanajuato DICIS, carr. Salamanca-Valle km. 3.5+1.8, Com. Palo Blanco, 36700 Salamanca, Mexico E-mail: palvarado@laviria.org, {ayalav,sanchezy}@ugto.mx

Several approaches using Evolutionary Algorithms (EA) can be found in the literature. The main variants of the EA for polygonal approximation of digital curves found in these methods include: 1) Modifications to the genetic operators of the EA. For example, Tsai [3] modifies the selection process to favor individuals according to its probability of being a vertex of the optimal shape. 2) Hybridization of several evolutionary techniques. We can find several combinations of GA, PSO and ACO in the literature. Yin shows in [4] that an hybrid technique of PSO and AG over performs methods using a single evolutionary approach. 3) Use of multi-objective algorithms, as those presented by Locteau *et. al.* [5] and Ho *et. al.* [6].

Different approaches optimize different criteria. For example, Traver *et. al.* [7] propose to optimize the length of the segments, the number of dominant points and the error approximation. Tsai [3] considers as the optimization criteria the orthogonal distance from the points in the shape encoded by the individual to the points in the digital curve to be approximated. Guanghui and Chuanbo [8] use a fitness functions composed of two error terms: RISE (Revised Internal Square Error) and BISE (Balance Internal Square Error). The optimization criteria considered for these terms are local sum of the square error, the number of segments and the length of the segments.

For all the evolutionary optimization approaches reviewed here, the individual representing the approximated shape is encoded using a string of N bits, where N is the number of total points in the reference figure. Each bit is encoded as a “1” if the corresponding point is chosen as a vertex and a “0" otherwise. That is, a notion of vertex order is given as input to the algorithms.

OUR APPROACH

We present an evolutionary programming approach (EP) [9] to solve the polygonal approximation of digital curves. The proposed method works both on open and closed digital curves without needing any *a priori* notion of vertex ordering. The approach proposed here can also estimate the number of vertices to approximate the reference curve. We use as optimization criteria six factors explained later to approximate a broader extent of digital curves.

METHODOLOGY

Problem Formulation

The algorithm receives a contour binary image as input. We consider that boundary is formed by N_E points (pixels) whose coordinates (x, y) are stored in a vector V. This method is proposed by Ayala *et al.* in [10] and [11]. In this form, it is not necessary to specify the order of the points of the reference figure. The objective of the algorithm is to find a set of N_P points belonging to V that can be used as vertices, which linked consecutively result in the best approximation to the reference shape.

We have used EP to find the set of vertices that best approximate the reference figure, including the estimation of the optimal number of them. The choice of EP was justified because its simplicity to handle variable length chromosomes [12]. In EP, we apply only mutation and replacement genetic operators, avoiding the need of crossover operations.

As pointed out by [9] [12], the main steps of EP are shown in the flowchart in figure 1.

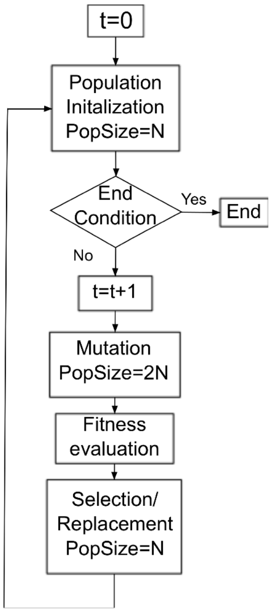

Figure 1. Main steps of EP.

As in other EAs, we need to define the individual representation, the fitness function, the mutation operator an a replacement (selection) mechanism. In the following sections, we describe how these elements of our approach were implemented.

Individual Representation

Each individual I in the EP population is defined as a $N_P - tuple$ of integer indexes $i_k \in 1, ..., N_E$, with $k \in 1, ..., N_P$. A given index k corresponds to a point $p_k(x_k, y_k)$ stored in V.

Since each figure can be approximated with a different number of vertices N_V, the size of the individual (N_I) is variable. A maximum (N_{max}) and a minimum (N_{min}) number of points are defined, so $N_V \in N_{min}, ..., N_{max}$. We have set $N_{min} = 3$ and $N_{max} = 20$. If we use N_B bits to represent a segment, the size of the individual is then $N_I = N_V * N_B$ bits.

The encoded individual represents an ordered sequence of straight line segments corresponding to a feasible solution of the digital curve approximation problem. The encoded solution uses the corresponding points in V as vertices of the piecewise linear approximation to the digital curve. The position of each vertex in the individual phenotype encodes the order of the linking of the line segments in the solution encoded by the individual.

Mutation Operator

Mutation serves in EP to evolve the population by generating new individuals from those belonging to the current population. Each individual in the EP population generates an offspring by using one of the five mutation procedures provided in our implementation. The proposed mutation operators M_i $i \in 1, ..., 5$ are described as follows:

Mutation on n bits (M_1). The new individual conserves the bit length of its parent but n bits are mutated. The bits to be mutated are chosen randomly.

Add a point at the end (M_2). This mutation operator adds a randomly chosen segment to the end of the bit string of its parent. So the bit string length is augmented by N_B bits

Remove a point at the end (M_3). The offspring is generated by deleting the ending point of the parent individual. The bit string length of the offspring is reduced N_B bits with respect to that of the parent.

Insert a point (M_4). The mutated individual is a modified copy of the parent where one of the segments of the parent is broken into two separated segments that are joined at the inserted point. Bit string length of the offspring is N_B longer than that of the parent.

Remove a point (M_5). This mutation operator removes a point in a random position of the segment chain of the parent individual. After that, it links previous point in the chain to the following point into one segment. Consequently the mutated individual bit string length is reduced by N_B bits.

In order to choose which mutation operator to apply for each individual, we use a roulette-wheel approach. That is, we assign to each mutation operator M_i an *a priori* probability P_i representing our expertise for solving the polygonal approximation problem. According to this, we have assigned to M_1 the largest probability of the mutation operator because it is the operator that helps the most to conserve diversity in the population. The probabilities for all the mutation operators are presented in table 1.

Table 1 .

Operator probabilities.

Operator Mutation	P_i
M_1	0.32
M_2	0.17
M_3	0.17
M_4	0.17
M_5	0.17

Fitness Function

The fitness is composed by six measures which try to ensure that two main characteristics are exhibited by the individuals: *i)* the existence of points of the synthesized polygonal curve in the figure being approximated; and *ii)* the coverage of the entire digital curve through the synthesized polygonal approximation. These measures are described below:

1. **Overall Distance to Target Curve (F_1).** For each encoded solution, we consider the sum of distances from the line segments S_i with $i \in 2, ..., N_V - 1$, composing it, to the actual curve points in the image. In order to improve computation speed, we consider only a sample fraction of the points for each line segment.

2. **Non-collinearity Factor (F_2).** The non-collinearity factor let us to prevent the use of collinear straight line segments as consecutive segments of the polygonal approximation. We compute the angle between consecutive segments and we penalize the individual if this angle is near to 0, π and 2π radians up to an experimentally determined threshold U_1. In our experiments we have used $U_1 = \pi/18$.

3. **Vertex Over Crowding Factor (F_3).** The vertex overcrowding factor is used to penalize individuals that exhibit a large number of vertices in small regions. The radius of these regions is set to a threshold $U_2 = N_E/(N_P)^2$.

4. **Bounding Box Factor (F_4).** We use this factor to favor the coverage of the bounding box area

of the curve being approximated by the encoded solution.

5. **Length Similarity Factor (F_5).** This factor favors the encoded solution that exhibit a length close to the length of the curve being approximated.
6. **Recall Factor (F_6).** The recall factor $F_6(I)$ measures the number of points in the reference curve that are present in the encoded solution or in a given tolerance radius T. We have set $T = 2$.

The overall fitness function is then:

$$F = (1 + k_6F_6)(k_1F_1 + k_2F_2 + k_3F_3 + k_5F_5) + k_4F_4 + k_7F_6 \quad (1)$$

where $k_i \in 1, ..., 7$ were calculated automatically by using a Genetic Algorithm on a parameter tuning test. A circle image was chosen as the reference figure because it was a base case along the design process. The values of these parameters are presented in table 2.

Table 2 .

Weighting parameters for the fitness function.

Parameter	Value	Parameter	Value
k_1	2.00	k_5	4.35
k_2	47.31	k_6	36.46
k_3	20.38	k_7	3.91
k_4	0.24		

REPLACEMENT MECHANISM

After the mutation step, the EP population was doubled in size. We have ranked the entire population using as criteria its fitness score. The best half was conserved and the worst half was discarded. In this way, the EA exhibited a steady state policy, and as a worst case, the current best solution is kept. The entire EP process is iterated until the ending condition is fulfilled.

TEST PROTOCOL

The figures of our interest can be described with the use of two main qualitative features (see figure 2): its openess and its straightness . By openess, we mean the property of the starting point and the ending point being far from each other. So, in figure 2, we can say that curves 1, 4 and 7 exhibit the lower openess because they are closed and curves 3, 6 and 9 presents a larger openess property. We consider as straightness, the property of a curve of being composed exclusively of straight line segments. In figure 2, curves 1, 2 and 3 exhibit the largest straightness value whilst shapes 7, 8 and 9 are only composed by smooth curves. Our shape dataset is composed of 9 curves that cover a range of the qualitative properties.

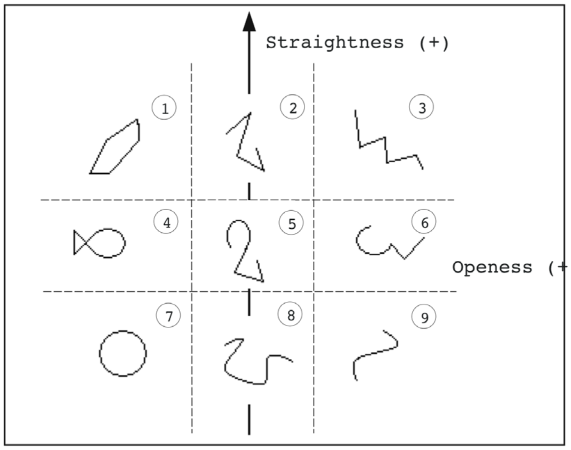

Figure 2 . Reference figures used to the tests performance.

Tuning Tests

In order to perform the tests with the aforementioned dataset, it was necessary to determine the number of individuals (N_{ind}) and the number of generations (N_{gen}) to be used in the algorithm implementation. For this purpose, we have tuned the algorithm by using as input figure exhibiting a mid range value for both properties of interest.

The reference curve for the tuning experiments is shown in figure 3. As we can observe there, the curve us composed of a mixture of curved segments and straight segments. The configuration cases considered in this tuning test were all the combinations in $N_{ind} \times N_{gen} = 200, 400, 600 \times 200, 400, 600$

Table 3 .

Tuning Parameters

N_{ind}	N_{gen}	Fitness
200	200	28774.45
	400	23907.81
	600	22500.44
400	200	1710.01
	400	1573.54
	600	1262.93
600	200	427.47
	400	190.18
	600	195.56

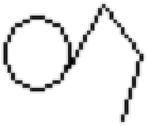

Figure 3 . Reference figure used to tune the algorithm.

The results of the optimal fitness for these tuning test is shown in table 3. The optimal configuration from the cases under study was achieved with $N_{ind} = 600$ and $N_{gen} = 400$.

RESULTS

Quantitative results for the performance test on the curves in figure 2 are shown in table 4. Average fitness ($\overline{F}$) and average number of points ($\overline{N_P}$) results for each digital curve are presented in columns two and three, respectively. Information about the best cases is also presented, fitness values (F_{Best}) and average distance error (E_{Best}) are shown in the last two columns. It is important to point out that average distance error is in pixels, and it was calculated on the resulting figure by using Bressenham's algorithm.

Table 4 .

Quantitative Results.

Figure	$\overline{F}$	$\overline{N_V}$	F_{Best}	E_{Best}
1	104.41	8.48	50.72	0.33
2	113.48	9.28	42.24	0.26
3	69.33	8.69	6.16	0.55
4	148.42	12.66	108.67	0.56
5	221.92	12.82	50.30	0.30
6	351.72	11.82	121.40	0.60
7	142.11	13.56	104.01	0.53
8	214.10	12.38	82.26	0.65
9	40.76	7.91	14.02	0.82

In figure 4, we show qualitative results of the best approximations. The optimal number of points found for each figure is indicated in a box.

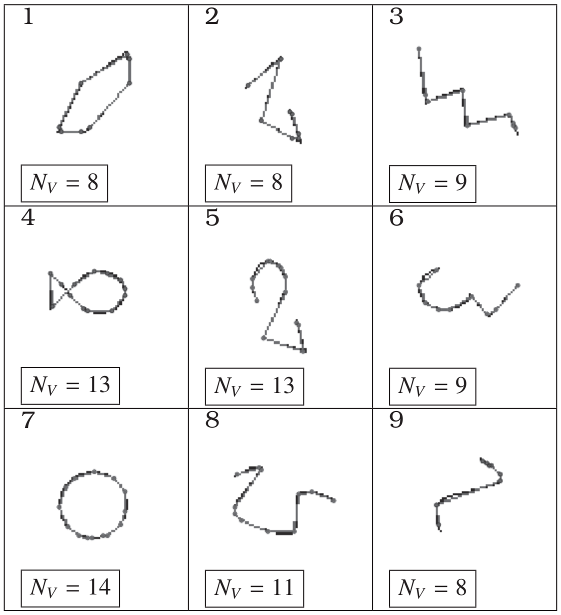

Figure 4 . Qualitative results of our method on a custom dataset. N_V represents the number of vertices.

With respect to the number of points in the resulting polygonal approximations, it is possible to say that figures with a high number of points are those which contain large curved zones, for example, curves 4, 5, 7 and 8. If there is not a defined number of vertices in the reference figure, a better approximation can be obtained by increasing the number of points on it. In curves where there is a defined number of vertices we could expect to obtain the same number but it does not happen. That can be explained by the individual fitness evaluation process, where a discretization process is done in order to reduce the computational time by avoiding to draw the figure with Bressenham's algorithm each time.

The best fitness values are obtained in figures with high straightness. Fitness values drops faster in this type of figures because it is easier to approximate straight lines than curved lines, therefore the individual evolutions gets also faster.

Finally, the average distance error take into account both fitness and number of points. Smaller errors will be obtained in figures with high straightness and/or high number of points. Even when the number of points in the resulting figure exceeds the number of points in the reference figure, the average distance error is not affected negatively.

CONCLUSIONS

A method to obtain a polygonal approximation of open and closed 2D figures with minimal *a priori* information was presented. The proposed method is based on the Evolutionary Programming approach which includes only mutation and replacement process. This evolutionary technique enables the individuals to be flexible with respect to the number of points depending of the reference figure. At the same time, since its evolutionary nature, EP optimize the sequence of the set of points which will be used as vertices in the polygonal approximation. Good quantitative and qualitative results were observed, even when the number of vertices in the resulting figure did not match with reference figure.

Work to be done includes to perform tests with larger images as well as the increment of maximal number of points allowed in each figure. The automatic computation of the fitness function parameters merits further study. In particular, we are going to explore how the parameters change when using different reference curves or even a sot of them simultaneously.

ACKNOWLEDGMENT

The authors thank the Dirección de Apoyo a la Investigación y el Posgrado of the Universidad de Guanajuato because its financial support through the project 123/09 of the Convocatoria Institucional 2009.

Paola B. Alvarado-Velazco gratefully acknowledges Mexico's CONACYT for the financial support through the scholarship 290604/368399.

REFERENCES

[1] Yin, P. Y., (2006). Genetic particle swarm optimization for polygonal approximation of digital curves. *Image Processing, Analysis, Recognition and Understanding* 16(2):223--233.

[2] Dong, F., Xiao, R., Zhong, Y., (2007). An approach to polygonal approximation of digital curves based on discrete particle swarm algorithm. *Journal of Universal Computer Science* 13(10):1449--1461.

[3] Tsai, Y.-H., (2006). Fast polygonal approximation based on genetic algorithms. In *Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse* , pp. 322--326.

[4] Yin, P.-Y., (2007). *Vision Systems: Segmentation and Pattern Recognition* , chapter Polygonal Approximation of Digital Curves Using the State-of-the-art Metaheuristics, pages 451--464. I-Tech.

[5] Locteau, H., Raveaux, R., Adam, S., Lecourtier, Y., Héroux, P., Trupin, E., (2006). Polygonal approximation of digital curves using a multi-objective genetic algorithm. In *The 18th International Conference on Pattern Recognition* , pages 716--719.

[6] Ho, S.-Y., Chen, J.-H., Huang, M.-H., (2004). Inheritable genetic algorithm for biobjective 0/1 combinatorial optimization problems and its applications. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* . 34(1):609--620.

[7] Traver, V. J., Recatalá, G., Iñesta, J. M., (2000). Exploring the performance of genetic algorithms as polygonal approximators. In *15th International Conference on Pattern Recognition* , pages 766--769.

[8] Guanghui, L., Chuanbo, C., (2007). A new approach for polygonal approximation of shape contours using genetic algorithms. In *Second IEEE Conference on Industrial Electronics and Applications* , pages 763--768.

[9] Fogel, L. J., (1999). *Intelligence through simulated evolution: forty years of evolutionary programming* . John Wiley & Sons.

[10] Ayala-Ramirez, V., Garcia-Capulin, C. H., Perez-Garcia, A., Sanchez-Yanez, R. E., (2006). Circle detection on image using genetic algorithms. *Pattern Recognition Letters* 27:652--657.

[11] Ayala-Ramirez, V., Mota-Gutierrez, S. A., Sanchez-Yanez, R. E., (2011). Quadrilateral detection using genetic algortihms. *Computación y Sistemas* 15(2).

[12] Rechy-Ramírez, F., Mesa, H.-G. A., Mezura-Montes, E., Cruz-Ramírez, N., (2011). *Advances in Soft Computing* , volume 7095, chapter Times Series Discretization Using Evolutionary Programming, pages 225--234. Springer Berlin / Heidelberg.