

Genetic Algorithm Works for Vectoring Image Outlines of Generic Shapes

Misbah Irshad¹, Muhammad Sarfraz², Malik Zawwar Hussain¹

¹Department of Mathematics, University of the Punjab, Lahore, Pakistan; ²Department of Information Science, Kuwait University, Kuwait City, Kuwait.
Email: misbah1109@gmail.com, malikzawwar.math@pu.edu.pk, prof.m.sarfraz@gmail.com

Received May 7th, 2013; revised June 8th, 2013; accepted June 16th, 2013

Copyright © 2013 Misbah Irshad *et al.* This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT

This work proposes a scheme which helps digitizing hand printed and electronic planar objects or vectorizing the generic shapes. An evolutionary optimization technique namely Genetic Algorithm (GA) is used to solve the problem of curve fitting with a cubic spline function. GA works well for finding the optimal values of shape parameters in the description of the proposed cubic spline. The underlying scheme comprises of various phases including data of the image outlines, detection of corner points, using GA for optimal values of shape parameters, and fitting curve using cubic spline to the detected corner points.

Keywords: Spline; Approximation; Curve Fitting; Genetic Algorithm; Generic Shapes

1. Introduction

Fitting curves to the data extracted from generic planar shapes is the problem which is immensely worked on during last two decades. It still grabs the attention of researchers due to its applications in diverse fields and its demands in the industry. The process of vectorizing outlines of the images consists of several mathematical and computational phases and stages. This process aims to fit an optimal curve to the data extracted from the boundary of the image. Although many contributions in the literature [1-27] can be found in this area, there is still room for making more advancements and finding interactive approaches.

Least square fitting is common in optimization problems in which splines and higher order polynomials are used to approximate the data. One can see a cubic spline technique [1] with least square fitting. Squared distance minimization has been used on B-spline curves in [2]. It uses iterative process to achieve an optimal curve.

Instead of parametric form, implicit form of the polynomial is also used for this purpose. Implicit B-spline curves [3] are used to solve curve reconstruction problem by approximating the point clouds. It uses the heuristic of trust region algorithm. In [4,5], schemes were proposed for fitting implicitly defined algebraic spline curves and surfaces. This was achieved over the scattered data by simultaneously approximating points and associated

normal vectors.

In this paper, a soft computing technique namely Genetic Algorithm (GA) [6] is proposed to find the optimal spline curves to the data extracted from the boundaries of the generic images. This evolutionary technique incorporates the corner points from the outline of the input image. The detection of corner points is quite significant as it helps minimizing the time to achieve desired curve to the outline of the image. Curve fitting in this scheme is done by using cubic spline which contains two shape parameters in its description. Basic target is to find those values of the parameters which assure minimum error between detected boundary of the image and the fitted spline curve.

The paper is organized in a way that the first and second steps (outline estimation and corner detection) of the proposed scheme are described in Section 2. A generalized cubic spline curve scheme is given in Section 3. Genetic Algorithm is explained in Section 4, Section 5 discusses the proposed scheme which is demonstrated with examples in Section 6. Finally, the paper is concluded in Section 7.

2. Countour Extraction and Segmentation

First step in proposed scheme of vectorization of planar objects is to extract data from the boundary of the bitmap image or a generic shape. In this procedure, a bitmap image of the generic shape is used as an input. In order to

get the image, software like Paint and Adobe Photoshop can be used or some other appropriate way can be adopted. After saving the bitmap image to the system, the chain code method [7,8] is used to extract boundary of the image. Chain codes represent the direction of the image and help to attain the geometric data from outline of the image.

In the next step, the data extracted from the outline needs to be subdivided into smaller segments for curve fitting. For this purpose corner points or significant points are detected. Detection of these points is not an easy task as exactness of detected corners can only be judged by human eye and no other standard criterion exist. Then accuracy of any corner detection scheme can only be examined if the original corner positions are known. Generally corner detection can be defined as an approach which extracts the dominating features of an image and consequently helps deducing contents of the image. Plenty of corner detection schemes can be found in the literature [9-12]. In this paper, the scheme presented in [10] is used to divide the boundary into smaller segments. Each segment of the boundary consists of two consecutive corner points and the data points in between them. These corner points would be used for curve fitting.

3. Generalized Cubic Spline

Finding corner leads to subdivision of the data obtained by the boundary of the bitmap image into pieces. Each piece consists of two successive corner points and the data points in between them. Thus if there are m corner points F_1, \dots, F_m then there will be m pieces P_1, \dots, P_m . Each piece is treated separately and spline is fitted to it.

First piece consists of all the contour points in between F_1 and F_2 inclusive. Second piece contains all contour points in between F_2 and F_2 inclusive. Consequently, the mth piece includes all contour points between F_m and F_1 inclusive. In general, the ith piece contains all the data points between F_i and F_{i+1} inclusive.

3.1. Cubic Spline Interpolant

As a curve fitting technique, the algorithm proposed in Section 5 makes use of a generalized cubic spline method. This spline embodies a number of desirable features needed for an optimum solution. The curve-fitting method employed here seeks the cubic spline for the determination of good shape parameters in its description.

Cubic spline function [13] is used for fitting curves at corner points. Let $F_i, F_{i+1}, i \in \mathbb{Z}$ be the two corner points of ith piece. Also let D_i and D_{i+1} be the corresponding tangents at corner points. Then the cubic function, where v_i and w_i are shape parameters, is defined by:

$$P|_{[t_i, t_{i+1}]}(t) = F_i(1-t)^3 + 3V_i(1-t)^2 t + 3W_i(1-t)t^2 + F_{i+1}t^3 \tag{1}$$

where

$$V_i = F_i + \frac{v_i h_i D_i}{3} \tag{2}$$

$$W_i = F_{i+1} - \frac{w_i h_i D_{i+1}}{3} \tag{3}$$

$$h_i = t_{i+1} - t_i > 0$$

Equation (1) can be rewritten as:

$$P|_{(t_i, t_{i+1})}(t) = R_{0,i}(t)F_i + R_{1,i}(t)V_i + R_{2,i}(t)W_i + R_{3,i}(t)F_{i+1} \tag{4}$$

where

$$\left. \begin{aligned} R_{0,i}(t) &= (1-t)^3, \\ R_{1,i}(t) &= 3t(1-t)^2, \\ R_{2,i}(t) &= 3t^2(1-t), \\ R_{3,i}(t) &= t^3. \end{aligned} \right\} \tag{5}$$

The functions $R_{j,i}, j = 0, 1, 2, 3$ are Bernstein Bézier like basis functions, such that

$$\sum_{j=0}^3 R_{j,i}(t) = 1 \tag{6}$$

The cubic function (1) has the following properties:

$$P(t_i) = F_i, P(t_{i+1}) = F_{i+1},$$

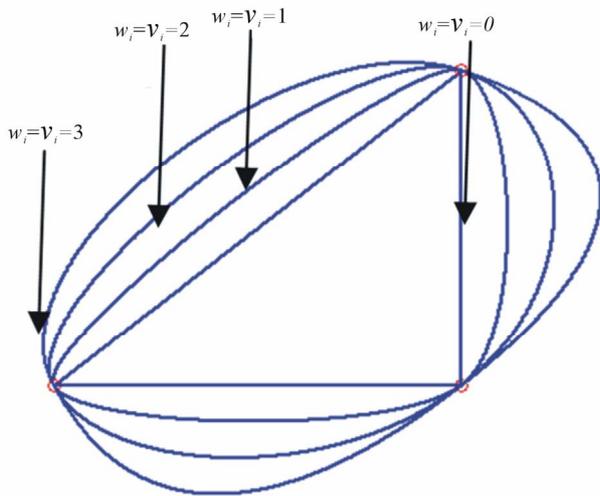
$$P^{(1)}(t_i) = v_i D_i, \text{ and } P^{(1)}(t_{i+1}) = w_i D_{i+1}, i \in \mathbb{Z}.$$

Figure 1 represents curve fitting to the given data by using cubic function (1) for assigning the different values to the parameters v_i and w_i . The effect of different values of the shape parameters on the shape of the curve are also shown in **Figure 1**. In **Figure 1(a)** cubic curve (1) is fitted to the data with the values of parameters as: $v_i = w_i = 0, v_i = w_i = 1, v_i = w_i = 2$ and $v_i = w_i = 3$. **Figures 1(b)** and **1(c)** show cubic curves with parameters $v_i = 3, w_i = 1$ and $v_i = 1, w_i = 3$ respectively.

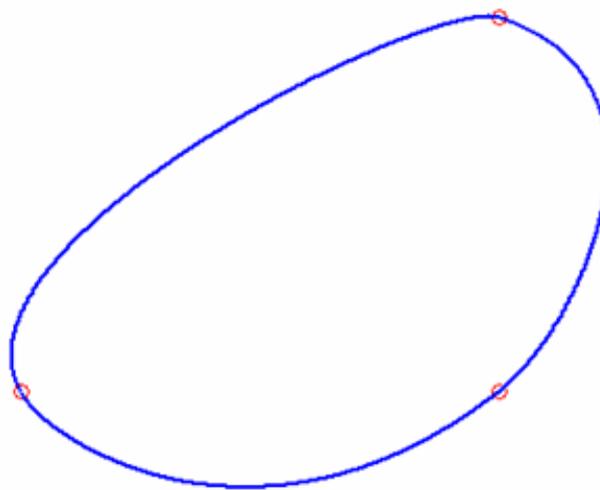
3.2. Parameterization

Number of parameterization techniques can be found in literature for instance uniform parameterization, linear or chord length parameterization, parabolic parameterization and cubic parameterization. In this paper, chord length parameterization is used to estimate the parametric value t associated with each point. It is as follows:

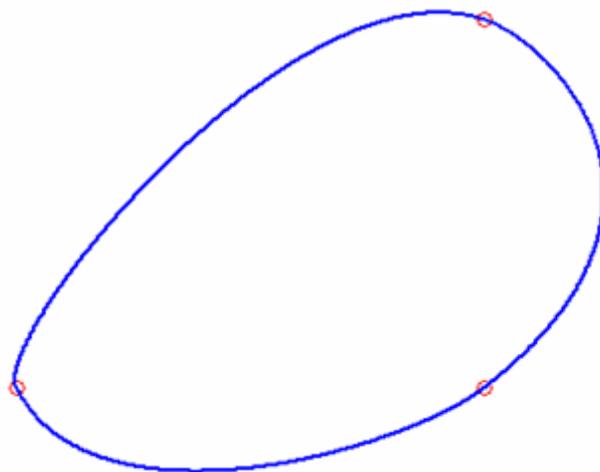
$$t = \begin{cases} 0 & \text{if } i = 1 \\ \frac{|pp| + |pp| + \dots + |pp|}{|pp| + |pp| + \dots + |pp|} & \text{if } 2 \leq i \leq n-1 \\ 1 & \text{if } i = n \end{cases}$$



(a)



(b): $w_i = 1$ and $v_i = 3$



(c): $w_i = 3$ and $v_i = 1$

Figure 1. Demonstration of cubic function (1) for different values of parameters.

It can be observed that t_i is in normalized form and varies from 0 to 1. Consequently, in our case, h_i is always equal to 1.

3.3. Estimation of Tangent Vectors

A distance based choice of tangent vectors D_i 's at F_i 's is defined as:

For open curves:

$$\left. \begin{aligned} D_0 &= 2(F_1 - F_0) - (F_2 - F_0)/2, \\ D_n &= 2(F_n - F_{n-1}) - (F_n - F_{n-2})/2, \\ D_i &= a_i(F_i - F_{i-1}) - (1 - a_i)(F_{i+1} - F_i), \\ & i = 1, 2, \dots, n-1. \end{aligned} \right\} \quad (7)$$

For close curves:

$$\left. \begin{aligned} F_{-1} &= F_{n-1}, F_{n+1} = F_1, \\ D_i &= a_i(F_i - F_{i-1}) - (1 - a_i)(F_{i+1} - F_i), \\ & i = 0, 1, \dots, n. \end{aligned} \right\} \quad (8)$$

where

$$a_i = \frac{|F_{i+1} - F_i|}{|F_{i+1} - F_i| + |F_i - F_{i-1}|}, \quad i = 0, 1, \dots, n. \quad (9)$$

4. Genetic Algorithm

Genetic Algorithms (GAs) are the evolution based search techniques. In GAs, every solution, in a given well-defined search space, is represented by a bit string. This bit string is called a chromosome. Selection, crossover and mutation are the three operators used in a genetic algorithm. A GA creates a population of chromosomes iteratively and is attempted to improve on the quality of chromosomes.

A GA allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the "fitness" (*i.e.*, minimizes the cost function). A set of input variable, in the form of a chromosome solution, is represented in a well-defined search space. A cost function, which may be a game, or an experiment or a mathematical function, is used to generate an optimal output from the chromosome.

The GA begins by defining a chromosome or an array of variable values to be optimized. The variable values are represented in binary form, so the binary GA works with bits. However, the cost function normally needs continuous variable to use in its description. Therefore, the chromosome is decoded whenever the cost function is evaluated.

How, a chromosome is encoded in binary form, is shown in **Figure 2**.

The GA starts with a group of chromosomes known as the population. Next the variables are passed to the cost

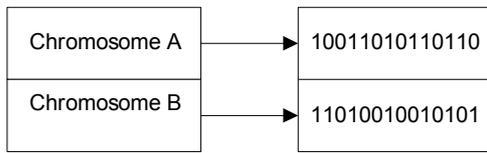


Figure 2. Example of binary encoding.

function for evaluation. Natural Selection process leads to Survival of the fittest *i.e.* discarding the chromosomes with the highest cost. Natural selection occurs in each generation or iteration of the algorithm. It is somewhat arbitrary to discard the undesired chromosomes or to keep the desired ones. If only very few chromosomes are allowed to survive for the next generation, it limits the available genes in the offspring. Similarly, if too many chromosomes are allowed to stay for next generation, the bad performers get a chance to contribute to the next generation in a bad way. Therefore, to have a natural selection process, it is recommended to keep 50% of the chromosomes.

Thresholding is another approach to the process of natural selection. All the chromosomes having a cost value less than some threshold are assumed to be survived in this approach. In order that parents produce offspring, the threshold allows some of the chromosomes to continue. Otherwise, to find some chromosomes that pass the test, there would be the case that the whole new population would be generated. In the whole process, in the beginning, a small number of chromosomes may survive. However, in the generations afterwards, most of the chromosomes will survive provided the threshold is not changed.

In process of matchmaking, two chromosomes are selected from the mating pool of survived chromosomes to produce two new offspring. There are several schemes for parent selection like roulette wheel, tournament selection, random pairing etc. The next step after selecting parents is mating to create one or more offspring.

The crossover operator is a commonly used form of mating. It deals with two parents to produce two offspring. The first and the last bits of the parent's chromosomes are used to randomly select a crossover point. The left of the crossover point to the first offspring is passed the binary code of the first parent. In the same way, the left of the crossover point to the second offspring is passed the binary code of the second parent. Moreover, the binary code to the right of the crossover point of first parent goes to second offspring and second parent passes its right side's code to first offspring. As a result of crossover operator the offspring contain parts of both the parents. Crossover operator is demonstrated in Figure 3.

Another way of creating new chromosomes is mutation in which new traits can be introduced to chromosomes that are not present in the original population. A

single point mutation changes a 1 to a 0, and *vice versa* is shown in Figure 4.

The process of GA described is iterated and would be repeated until the achievement of best solution for the problem. Flowchart of GA is shown in Figure 5.

5. Proposed Approach

In this Section the proposed scheme to the curve fitting problem is described. It includes the phases of problem matching with Genetic Algorithm using cubic spline function, description of parameters used for GA and curve fitting.

5.1. Problem Mapping

In this section Genetic Algorithm formulation of the pro-

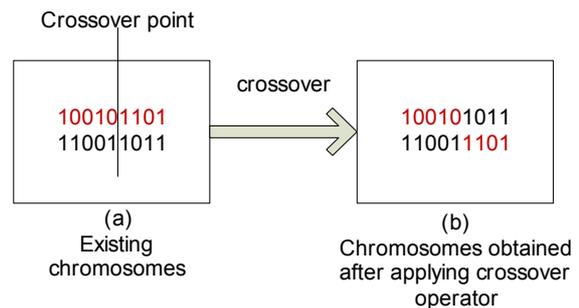


Figure 3. Example of crossover operator.

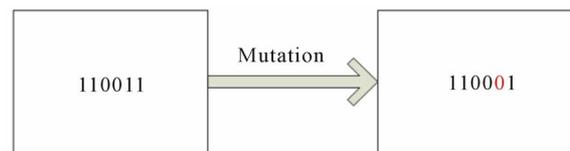


Figure 4. Example of mutation operator.

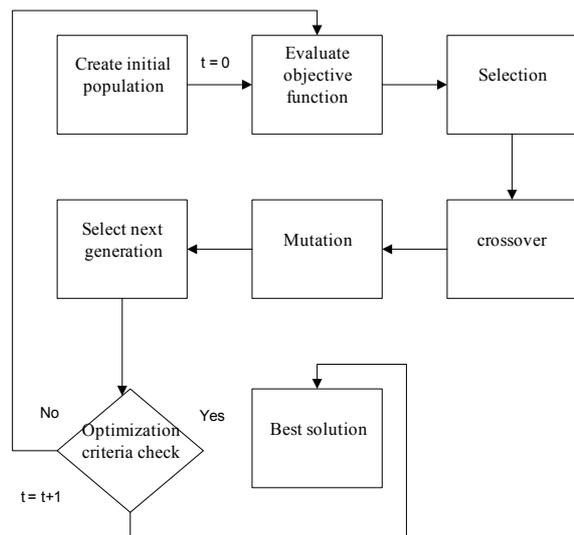


Figure 5. Flow diagram of genetic algorithm.

blem discussed in this paper is described in detail.

Suppose, for $i = 0, 1, \dots, n - 1$, the data segments $P_{i,j} = (X_{i,j}, Y_{i,j}), j = 1, 2, \dots, m$ are given as ordered sets of the universal set of data points. Then the squared sums S_i 's of distance between $P_{i,j}$'s and their corresponding parametric points $P(t_i)$'s on the curve are determined as

$$S_i = \sum_{j=1}^{m_i} [P_i(u_{i,j}) - P_{i,j}]^2 \quad i = 0, 1, \dots, n - 1,$$

where u 's are parameterized in reference to chord length parameterization. For the best fitting of the curve to given data, such values of parameter v_i and w_i , are required so that the sums S_i 's are minimal. Genetic Algorithm is used to optimize this value for the fitted curve. We start with initial population of values of v_i and w_i chosen randomly. Successive application of search operations to this population leads to optimal values of v_i and w_i .

5.2. Initialization

Once we have the bitmap image shown in **Figures 6(a), 7(a) and 8(a)**, the method of Section 2 is used to extract the boundary of the image. The boundary of the image is then used to detect the corner points in the next phase. It uses the corner detection method pointed out in Section 2. **Figures 6(b) and 6(c), Figures 7(b) and 7(c) and Figures 8(b) and 8(c)**, show boundary of the bitmap images and detected corner points respectively. **Table 1** gives number of contour points and initial corner points of the images.

5.3. Curve Fitting

Detection of the corner points leads towards the subdivision of the boundary of the image into segments. The interpolation spline of Section 3.1 is then used to approximate each segment of the boundary. This spline has the parameters v and w in its description. The initial solution of the parameters v and w is randomly selected. After an initial approximation for the segment is obtained, The GA is run to get the optimal solution of v and w . Genetic Algorithm helps to obtain better approximations to achieve optimal solution. The tangent vectors at knots are estimated by the method described in Section 3.3.

5.4. Breaking Segment

For some segments, the best fit obtained through iterative improvement may not be satisfactory. In that case, we subdivide the segment into smaller segments at points where the distance between the boundary and parametric curve exceeds some predefined threshold; such points are termed as intermediate points. A new parametric curve is fitted for each new segment as shown in **Figures 6(e)**

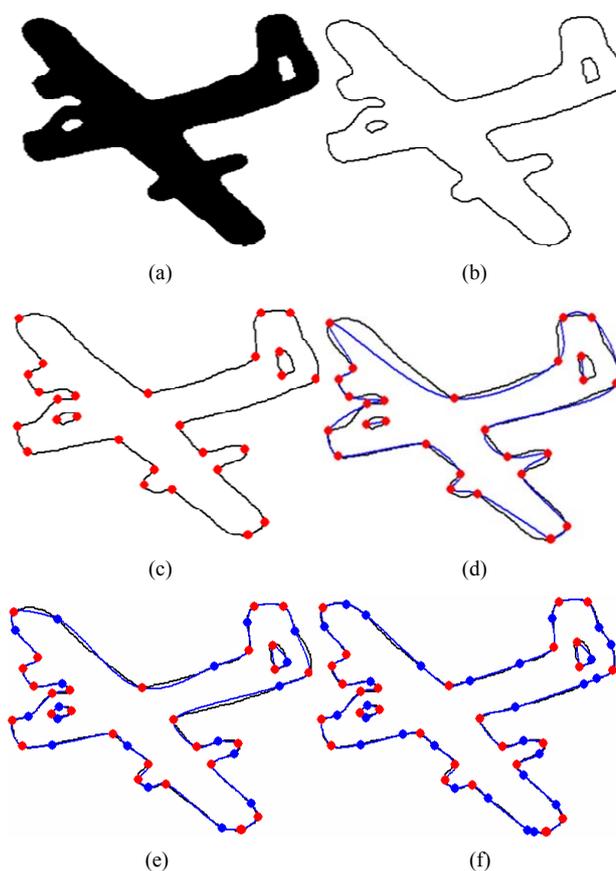


Figure 6. Image of Plane with detected corner points and fitted cubic curve. (a) Bitmap image; (b) Boundary extracted; (c) Corners detected; (d) Cubic curve interpolated to corner points for 1st iteration of GA; (e) Cubic curve interpolated to corner points for 2nd iteration of GA with breakpoints; (f) Cubic curve interpolated to corner points for final (5th) iteration of GA with breakpoints.

Table 1. Details of digital contours and corner points.

Image	Name	#of contours	#of contour points	#of initial corner points
	Fork.bmp	1	673	15
	Plane.bmp	3	915 + 36 + 54	28
	Fish.bmp	1	975	32

and **6(f), Figures 7(e) and 7(f) and Figures 8(e) and 7(f)**. In **Table 2**, number of intermediate points is presented which is obtained while fitting the optimized cubic spline for different iterations of GA.

6. Demonstration

Curve fitting scheme, proposed in Section 5 has been implemented on different images. In **Figure 6**, (a) represents original image, (b) shows outline of the image, (c)

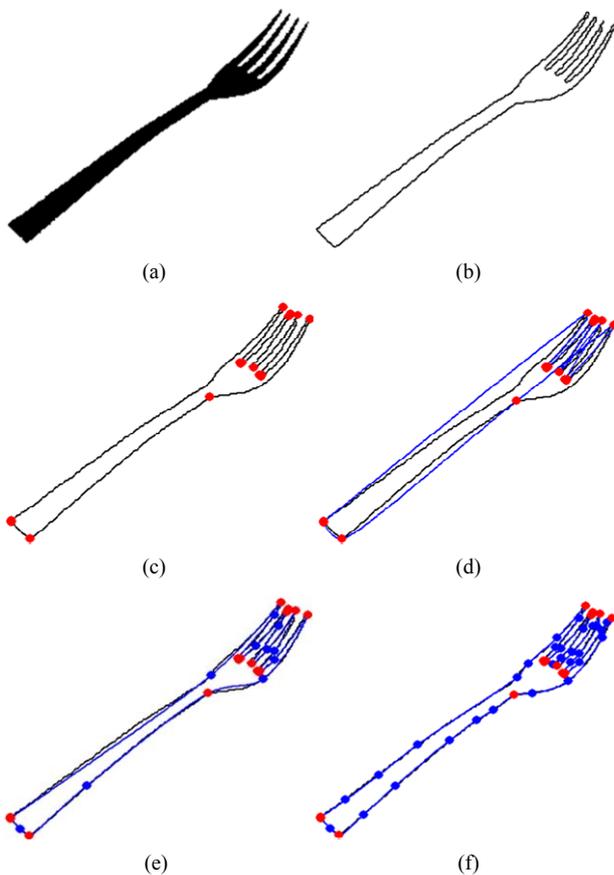


Figure 7. Image of fork with detected corner points and fitted cubic curve. (a) Bitmap image; (b) Boundary extracted; (c) Corners detected; (d) Cubic curve interpolated to corner points for 1st iteration of GA; (e) Cubic curve interpolated to corner points for 2nd iteration of GA with breakpoints; (f) Cubic curve interpolated to corner points for final (4th) iteration of GA with breakpoints.

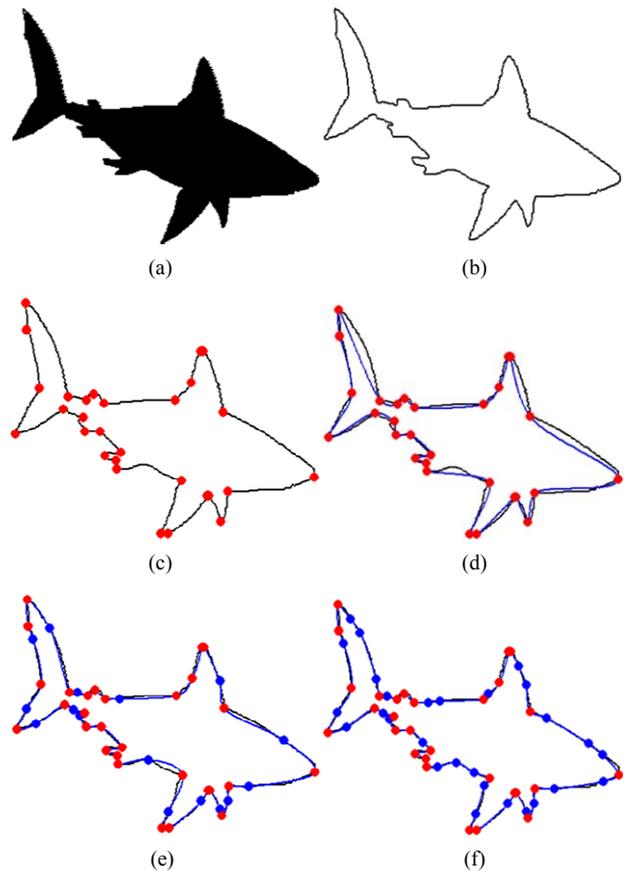


Figure 8. Image of Fish with detected corner points and fitted cubic curve. (a) Bitmap image; (b) Boundary extracted; (c) Corners detected; (d) Cubic curve interpolated to corner points for 1st iteration of GA; (e) Cubic curve interpolated to corner points for 2nd iteration of GA with breakpoints; (f) Cubic curve interpolated to corner points for final (4th) iteration of GA with breakpoints.

Table 2. Number of corner points together with number of intermediate points for iterations 1, 2 and final iteration of GA along with total time elapse in running GA.

Image Name	Number of initial corner points	Number of intermediate points in cubic interpolation with threshold value 3			Time Elapsed
		Itr.1	Itr.2	Final iteration	
Fork.bmp	15	23	27	31	0.453000 sec.
Plane.bmp	28	33	38	40	0.609000 sec.
Fish.bmp	32	25	29	31	0.625000 sec.

demonstrates corner points (d), (e) and (f) give fitted outline for 1st , 2nd and final iterations for threshold 3 respectively using Genetic Algorithms together with corner points and intermediate points). Figures 7 and 8 can also be described in similar fashion. Time elapsed for applying the proposed scheme for different images is given in Table 2.

Figures 9-13 show behaviors of fitness function for the image of fish on running GA again and again. It can be observed in Figure 9 that minimum value of cost function is achieved after iteration 20, whereas Figure 10 and Figure 11 indicate that minimum fitness function is obtained at iteration 10 and iteration 5 respectively. While Figures 12 and 13 depict a bit different behavior as in these cases initially fitness function increases and then it starts decreasing.

In Figure 14, stopping criteria followed to run GA is given and in Figure 15 best (^), worst (o) and mean (*) values of objective functions are shown in each iteration for the image of fish. Flowchart for proposed algorithm is given in Figure 16.

7. Conclusion

In this paper a scheme is presented which vectorizes the generic shapes. A cubic function is used for curve fitting and a soft computing technique genetic algorithm is used

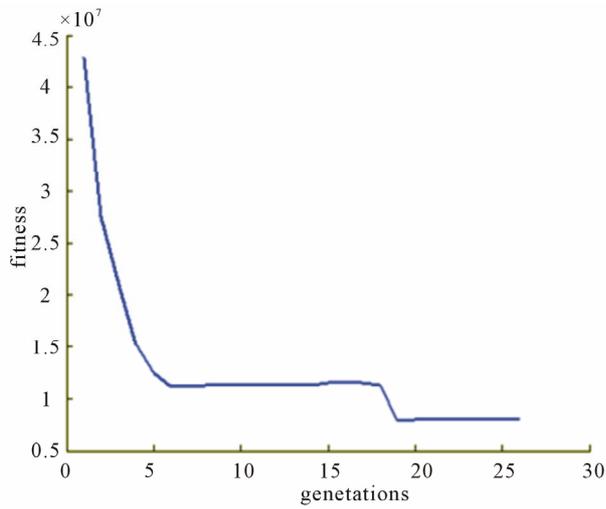


Figure 9. Graph of fitness function.

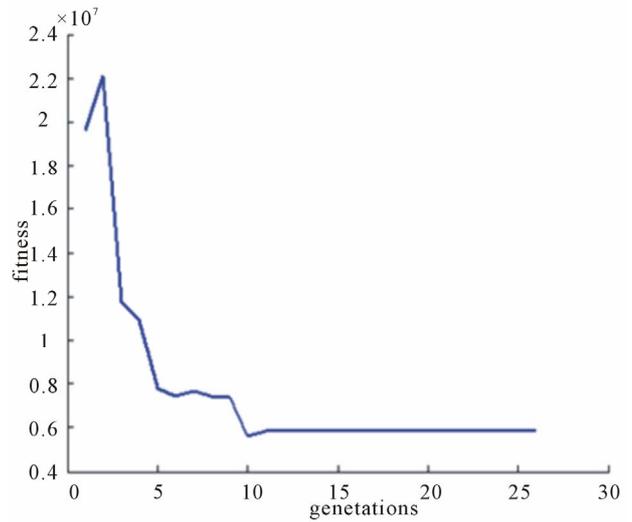


Figure 12. Mix behavior of fitness function.

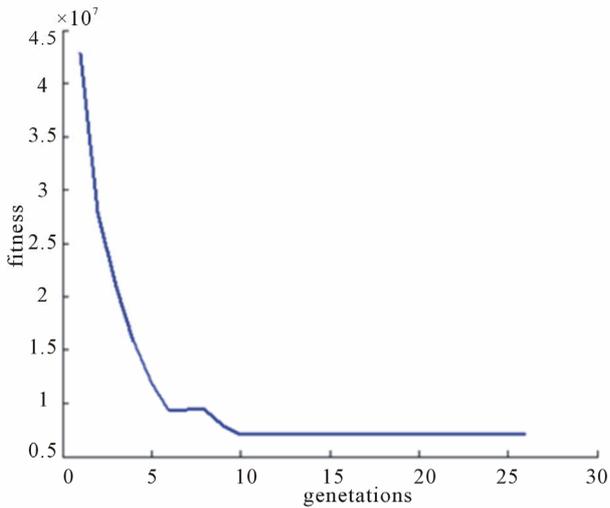


Figure 10. Behavior of fitness function.

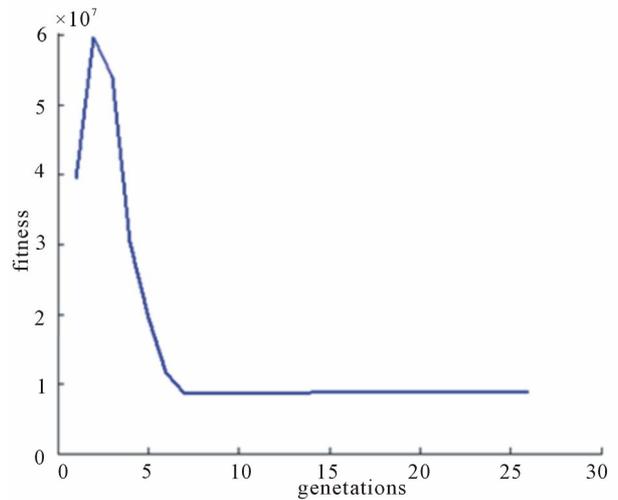


Figure 13. Increasing and decreasing fitness function.

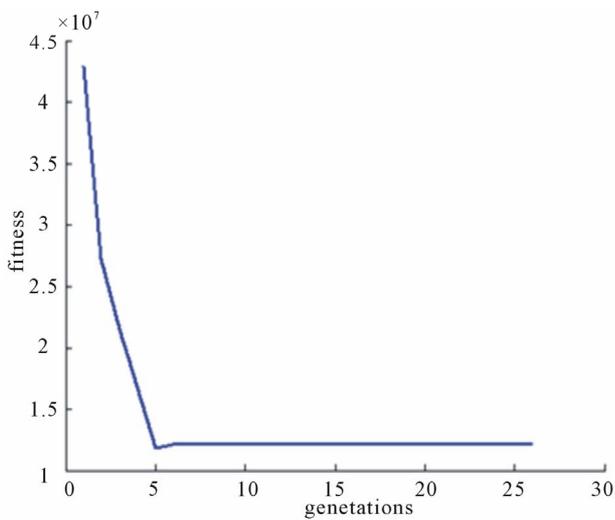


Figure 11. Behavior of fitness function.

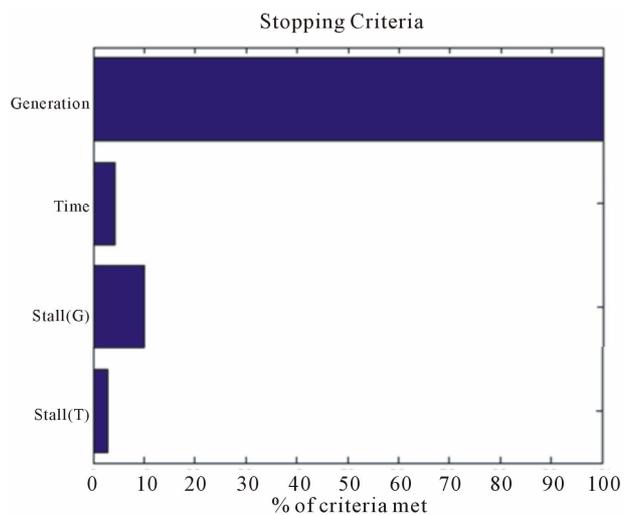


Figure 14. Stopping criteria met by GA in %.

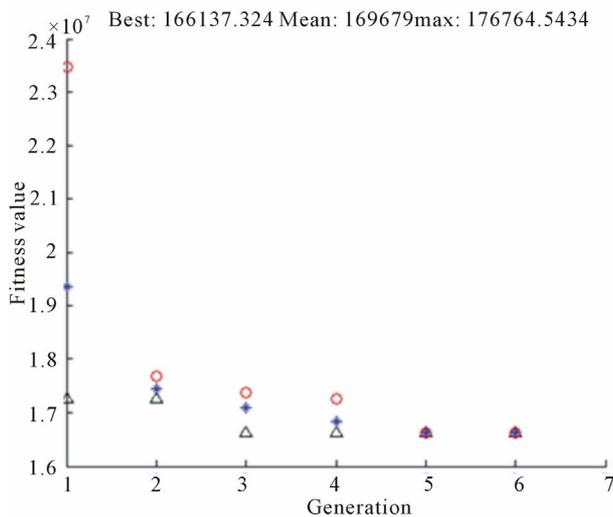


Figure 15. Best, worst and mean scores in different iterations.

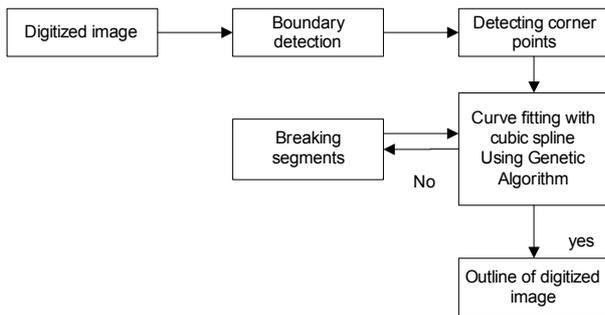


Figure 16. Outline of proposed algorithm.

to find optimal values of the parameters in the description of the cubic function. The method proposed starts with initial random population of parameters and finds those values of the parameters which can assure best optimal curve to the data extracted by bitmap images. The scheme presented is automatic and no human intercession is required. It also ensures computational efficiency as far as curve fitting is concerned.

REFERENCES

- [1] M. Sarfraz and M. A. Khan, "An Automatic Algorithm for Approximating Boundary of Bitmap Characters," *Future Generation Computer Systems*, Vol. 20, No. 8, 2004, pp. 1327-1336. [doi10.1016/j.future.2004.05.024](https://doi.org/10.1016/j.future.2004.05.024)
- [2] X. Yang, "Curve Fitting and Fairing Using Conic Splines," *Computer Aided Design*, Vol. 36, No. 5, 2004, pp. 461-472. [doi10.1016/S0010-4485\(03\)00119-2](https://doi.org/10.1016/S0010-4485(03)00119-2)
- [3] G. Lavoue, F. Dupont and A. Baskurt, "A New Subdivision Based Approach for Piecewise Smooth Approximation of 3D Polygonal Curves," *Pattern Recognition*, Vol. 38, No. 8, 2005, pp. 1139-1151. [doi10.1016/j.patcog.2005.02.002](https://doi.org/10.1016/j.patcog.2005.02.002)
- [4] B. S. Morse, T. S. Yoo, D. T. Chen, P. Rheingans and K. R. Subramanian, "Interpolating Implicit Surfaces from Scattered Surface Data Using Compactly Supported Radial Basis Functions," *Proceedings of Conference on Shape Modeling and Applications*, Genova, 7-11 May 2001, pp. 89-98. [doi10.1109/SMA.2001.923379](https://doi.org/10.1109/SMA.2001.923379)
- [5] X. N. Yang and G. Z. Wang, "Planar Point Set Fairing and Fitting by Arc Splines," *Computer Aided Design*, Vol. 33, No. 1, 2001, pp. 35-43. [doi10.1016/S0010-4485\(00\)00059-2](https://doi.org/10.1016/S0010-4485(00)00059-2)
- [6] D. E. Goldberg, "Geneticalgorithms in Search, Optimization and Machine Learning," Addison-Wesley, Reading, 1989.
- [7] G. Avrahami and V. Pratt, "Sub-Pixel Edge Detection in Character Digitization," *Proceedings of International Conference on Raster Imaging and Digital Typography II*, Boston, December 1991, pp. 54-64.
- [8] Z. J. Hou and G. W. Wei, "A New Approach to Edge Detection," *Pattern Recognition*, Vol. 35, No. 7, 2002, pp. 1559-1570. [doi10.1016/S0031-3203\(01\)00147-9](https://doi.org/10.1016/S0031-3203(01)00147-9)
- [9] H. L. Beus and S. S. H. Tiu, "An Improved Corner Detection Algorithm Based on Chain Coded Plane Curves," *Pattern Recognition*, Vol. 20, No. 3, 1987, pp. 291-296. [doi10.1016/0031-3203\(87\)90004-5](https://doi.org/10.1016/0031-3203(87)90004-5)
- [10] D. Chetrikov and S. Zsabo, "A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves," *Proceedings of the 23rd Workshop of Austrian Pattern Recognition Group*, Steyr, 27-28 May 1999, pp. 175-184.
- [11] H. Freeman and L. S. Davis, "A Corner Finding Algorithm for Chain-Coded Curves," *IEEE Transaction on Computer*, Vol. 26, No. 3, 1977, pp. 297-303. [doi10.1109/TC.1977.1674825](https://doi.org/10.1109/TC.1977.1674825)
- [12] B. Jüttler and A. Felis, "Least Square Fitting of Algebraic Spline Surfaces," *Advances in Computational Mathematics*, Vol. 17, No. 1-2, 2002, pp. 135-152. [doi10.1023/A:1015200504295](https://doi.org/10.1023/A:1015200504295)
- [13] M. Sarfraz, M. Z. Hussain and F. S. Chaudary, "Shape Preserving Cubic Spline for Data Visualization," *Computer Graphics and CAD/CAM*, Vol. 1, No. 6, 2005, pp. 185-193.
- [14] T. Harada, F. Yoshimoto and Y. Aoyama, "Data Fitting Using a Genetic Algorithm with Real Number Genes," *Proceedings of the IASTED International Conference on Computer Graphics and Imaging*, Las Vegas, 1-7 November 2000, pp. 131-138.
- [15] J. H. Horng, "An Adaptive Smoothing Approach for Fitting Digital Planar Curves with Line Segments and Circular Arcs," *Pattern Recognition Letters*, Vol. 24, No. 1-3, 2003, pp. 565-577. [doi10.1016/S0167-8655\(02\)00277-5](https://doi.org/10.1016/S0167-8655(02)00277-5)
- [16] M. Jyoti, D. Ratna and G. Sainarayanan, "Harris Operator Corner Detection Using Sliding Window Method," *International Journal of Computer Applications*, Vol. 22, No. 1, 2011, pp. 28-37.
- [17] H. Kano, H. Nakata and C. F. Martin, "Optimal Curve Fitting and Smoothing Using Normalized Uniform B-Splines: A Tool for Studying Complex Systems," *Applied Mathematics and Computation*, Vol. 169, No. 1, 2005, pp.

- 96-128. [doi10.1016/j.amc.2004.10.034](https://doi.org/10.1016/j.amc.2004.10.034)
- [18] S. Kirkpatrick, C. D. Gelatt Jr. and M. P. Vecchi, "Optimization by Simulated Annealing," *Science, New Series*, Vol. 220, No. 4598, 1983, pp. 671-680.
- [19] M. Moriyama, F. Yoshimoto and T. Harada, "A Method of Plane Data Fitting with a Genetic Algorithm," *Proceeding of the IASTED International Conference on Computer Graphics and Imaging*, Ames, 11-12 May 1998, pp. 21-31.
- [20] M. Sarfraz and A. Raza, "Visualization of Data Using Genetic Algorithm," *Soft Computing and Industry, Recent Applications*, 2002, pp. 535-544.
- [21] M. Sarfraz, "Some Algorithms for Curve Design and Automatic Outline Capturing of Images," *International Journal of Image and Graphics*, Vol. 4, No. 2, 2004, pp. 301-324. [doi10.1142/S0219467804001427](https://doi.org/10.1142/S0219467804001427)
- [22] M. Sarfraz, "Computer-Aided Reverse Engineering Using Simulated Evolution on NURBS," *International Journal of Virtual and Physical Prototyping*, Vol. 1, No. 4, 2006, pp. 243-257. [doi10.1080/17452750601130492](https://doi.org/10.1080/17452750601130492)
- [23] M. Sarfraz and A. Rasheed, "A Randomized Knot Insertion Algorithm for Outline Capture of Planar Images Using Cubic Spline," *The Proceedings of the 22nd ACM Symposium on Applied Computing*, Seoul, 11-15 March 2007, pp. 71-75.
- [24] M. Sarfraz, "Vectorizing outlines of Generic shapes by Cubic Spline Using Simulated an Nealing," *International Journal of Computer Mathematics*, Vol. 87, No. 8, 2010, pp. 1736-1751. [doi10.1080/00207160802452519](https://doi.org/10.1080/00207160802452519)
- [25] W.-C. Hu, "Multiprimitive Segmentation Based on Meaningful Break Points for Fitting Digital Planar Curves with Line Segments and Conic Arcs," *Image and Vision Computing*, Vol. 23, No. 9, 2005, pp. 783-789. [doi10.1016/j.imavis.2005.05.004](https://doi.org/10.1016/j.imavis.2005.05.004)
- [26] H. Yang, W. Wang and J. Sun, "Control Point Adjustment for B-Spline Curve Approximation," *Computer Aided Design*, Vol. 36, No. 7, 2004, pp. 639-652. [doi10.1016/S0010-4485\(03\)00140-4](https://doi.org/10.1016/S0010-4485(03)00140-4)
- [27] Z. Yang, J. Deng and F. Chen, "Fitting Unorganized Point Clouds with Active Implicit B-Spline Curves," *Visual Computing*, Vol. 21, No. 1, 2005, pp. 831-839. [doi10.1007/s00371-005-0340-0](https://doi.org/10.1007/s00371-005-0340-0)