Sains Malaysiana 42(8)(2013): 1167–1179

Reverse Engineering of Planar Objects Using GAs
(Kejuruteraan Balikan Objek Menyatah Menggunakan GA)

MUHAMMAD SARFRAZ*, MISBAH IRSHAD & MALIK ZAWWAR HUSSAIN

ABSTRACT

An automatic approach, for reverse engineering of digitized hand printed and electronic planar objects, is presented which is useful for vectorizing the generic shapes. The rational cubic functions were used to find the optimal solution of the curve fitting problem with the help of a soft computing technique genetic algorithm which finds appropriate values of shape parameters in the description of rational cubic functions.

Keywords: Curve fitting; generic shapes; genetic algorithm; rational functions; reverse engineering

ABSTRAK

Kaedah automatik untuk kejuruteraan balikan bagi objek cetakan tangan terdigit dan elektronik dibentangkan. Kaedah ini berguna untuk mevektor bentuk generik. Fungsi kubik rasional digunakan untuk mencari penyelesaian optimum masalah pemadanan lengkung menggunakan teknik pengkomputeran algoritma genetik lembut yang mencari nilai yang sesuai bagi parameter bentuk untuk menerangkan fungsi rasional kubik.

Kata kunci: Algoritma genetik; bentuk generik; fungsi rasional; kejuruteraan balikan; pemadanan lengkung

INTRODUCTION

Reverse engineering, which deals with a number of activities, has been mounting immensely in the last decade. The reverse engineering procedure usually is to break apart some object or system in order to explore its technical principles and mechanism so that an improved or duplicate system can be developed, when no original mechanical drawing, documentation or computer model are existed. Generating computer aided design (CAD) model from scanned digital data is used in contour styling which needs to adopt some curve or surface approximation scheme.

Reverse engineering of 2D objects aims to fit an optimal curve to the data extracted from the boundary of the image (Hou & Wei 2002; Kirkpatrick et al. 1983; Sarfraz 2004; Sarfraz & Khan 2004; Sarfraz & Rasheed 2007). Fitting curves to the data extracted by generic planar shapes is the problem which is immensely worked on during the last two decades. It still grabs the attention of researchers due to its applications in diverse fields and its demands in the industry. There are several advantages of curved representation of planar objects, for example, transformations like scaling, shearing, translation, rotation and clipping can be applied on the objects very easily.

Curve fitting is vastly used in reverse engineering in order to reproduce curves from exact geometric data obtained by bitmap images; consequently it is always required to provide new curve-fitting algorithms. Various outline capturing techniques using different spline models have been introduced by researchers like Be´zier splines (Sarfraz & Rasheed 2007), B-splines (Ho¨ lzle 1983), Hermite interpolation (Sarfraz & Razzak 2002) and rational cubic interpolation (Sarfraz & Khan 2000, Sarfraz et al. 2012). There are several other outline capturing techniques (Cabrelli & Molter 1990; Davis 1979; Itoh & Ohno 1993; Plass & Stone 1983; Sarfraz 2004; Sarfraz et al. 2005; Sarfraz & Khan 2002, 2004; Sarfraz & Raza 2001, 2002; Schneider 1990; Sohel et al. 2005; Tang et al. 2001) available in the current literature and most of them are based on least-squares fit (Itoh & Ohno 1993; Plass & Stone 1983; Sarfraz & Khan 2002) and error minimization (Cabrelli & Molter 1990; Sarfraz & Khan 2000, 2004). Sarfraz et al. (2004) in their outline capturing scheme, calculated the ratio between two intermediate control points and used this to estimate their position. This caused reduction of computation in subsequent phases of approximation. Few other techniques include use of control parameters (Sarfraz & Razzak 2002), genetic algorithms (Sarfraz & Raza 2001) and wavelets (Tang et al. 2001). This paper is oriented to the rational spline method of Sarfraz et al. (2012) and extends the work in detail and elaborates in more extensive way.

The computing method, analogous to the amazing aptitude of the human mind to reason and learn in an environment of uncertainty and imprecision, is called soft computing. In an attempt to find out reasonably useful solutions, soft computing- based optimization methods acknowledge the presence of imprecision and uncertainty present in optimization. Soft computing techniques such as fuzzy logic (FL), neural networks (NN), genetic algorithm (GA), simulated annealing (SA), ant colony optimization (ACO) and particle swarm optimization (PSO) have received a lot of attention of researchers due to their potentials to deal with highly nonlinear, multidimensional and ill-behaved complex engineering problems (Chandrasekaran et al. 2010).

The genetic algorithm has been very powerful tool in finding out the global minima which is derived from the process of natural evolution based on the 'survival of the fittest' philosophy (Goldberg 1989). So it was used in this paper to find optimal curve to the data obtained from boundary of the image. The process of vectorizing outlines of the images consists of several mathematical and computational stages like: Extracting boundary of the bitmap image, detecting corners from the boundary and fitting optimal curve to the corners.

COUNTOUR EXTRACTION AND SEGMENTATION

The first step in reverse engineering of planar objects is to extract data from the boundary of the bitmap image or a generic shape. Capturing boundary or outline representation of an object is a way to preserve the complete shape of an object. The objects in an image can also be represented by the interior of shape. Chain coding for boundary approximation and encoding was initially proposed by (Freeman 1961), which has drawn significant attention over the last three decades. Chain codes represent the direction of the image and help to attain the geometric data from outline of the image. Extracted boundaries of the bitmap images given in Figures 7(a), 8(a) and 9(a) are in Figures 7(b), 8(b) and 9(b), respectively.

The second step in reverse engineering of planar objects is segmentation of object boundary before curve fitting, it is very important for two reasons. Firstly, it reduces boundary's complexity and simplifies the fitting process. Secondly, each shape consists of natural break points (like four corners of a rectangle) and quality of approximation can be improved if boundary is subdivided into smaller pieces at these points. These are normally the discontinuous points to which we do not want to apply any smoothing and like to capture them as such. These points can be determined by a suitable corner detector. Researchers have used various corner detection algorithms for outline capturing (Avrahami & Pratt 1991; Beus & Tiu 1987; Chetrikov & Zsabo 1999; Sarfraz et al. 2006; Teh & Chin 1989). The method proposed in (Chetrikov & Zsabo 1999) was used in this paper. Number of contour points and detected boundary points for different images is given in Table 1. Detected corners of the boundaries shown in Figures 7(b), 8(b) and 9(b) can be seen in Figures 7(c), 4(c) and 5(c), respectively.

RATIONAL CUBIC FUNCTION

A piecewise rational cubic parametric function $P \in C^1[t_i, t_{i+1}]$, with shape parameters v_i, $w_i \geq 0$, $i = 1, \ldots, n$ is used for curve fitting to the corner points detected from the boundary of the bitmap image, the rational cubic function is defined for $t \in [t_i, t_{i+1}]$, $i = 1, \ldots, n$, as follows:

$$P_i(t) = \frac{F_i(1-\theta)^3 + v_i V_i (1-\theta)^2\theta + w_i W_i (1-\theta)\theta^2 + F_{i+1}\theta^3}{(1-\theta)^3 + v_i(1-\theta)^2\theta + w_i(1-\theta)\theta^2 + \theta^3}, \quad (1)$$

where F_1 and F_{i+1} are two corner points (given control points) of the i^{th} segment of the boundary with $\theta = (t - t_i)/h_i$, wherte $h_i = t_{i+1} - t_i$,

$$V_i = F_i + \frac{h_i D_i}{v_i} \text{ and } W_i = F_{i+1} - \frac{h_i D_{i+1}}{w_i}, \quad (2)$$

where D_i, $i = 1, \ldots, n + 1$ are the first derivative values at the knots t_i, $i = 1, \ldots, n + 1$. The effect of the shape parameters v_i, $i = 1, \ldots, n$, on the curve is shown in Figures 1 and 2. Moreover, for $v_i, w_i = 3$, $i = 1, \ldots, n + 1$ (1) reduces to cubic Hermite interpolation. If $v_i, w_i \to \infty$ then the rational cubic function (1) converges to linear interpolant $L_i(t) = (1 - \theta) F_i + \theta F_{i+1}$ as shown in Figure 1. Furthermore it can be observed that the function (1) may have two sub cases as:

Case 1: $v_i = w_i$, $i = 1, \ldots, n$
Case 2: $v_i \neq w_i$, $i = 1, \ldots, n$.

In this paper both the cases were discussed for the curve fitting. For $v_i = w_i$, $i = 1, \ldots, n$ (1) can be written in the form:

$$P_i(t_i;v_i) = R_0(\theta;v_i)F_i + R_1(\theta;v_i)V_i + R_2(\theta;v_i) W_i + R_3(\theta;v_i)F_{i+1}, \quad (3)$$

where V_i and W_i are given in (2) and $R_j(\theta;v_i)$, $j = 0,1,2,3$ are rational Bernstein-Bezier weight functions such that $\sum_{j=0}^{3} R_j(\theta;v_i) = 1$.

Similarly for $v_i, w_i \neq 0$ (1) will become:

$$P_i(t_i;v_i,w_i) = R_0(\theta;v_i,w_i)F_i + R_1(\theta;v_i,w_i) V_i + R_2(\theta;v_i,w_i)F_{i+1}, \quad (4)$$

where $R_j(\theta;v_i,w_i)$, $j = 0,1,2,3$ are rational Bernstein-Bezier weight functions such that $\sum_{j=0}^{3} R_j(\theta;v_i,w_i) = 1$, V_i and W_i are given in (2).

PARAMETERIZATION

The 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:

$$t_i = \begin{cases} 0 & if\ i = 1 \\ \dfrac{|p_1p_2| + |p_2p_3| + \ldots + |p_ip_{i+1}|}{|p_1p_2| + |p_2p_3| + \ldots + |p_{n-1}p_n|} & if\ 2 \leq i \leq n-1. \\ 1 & if\ i = n \end{cases}$$

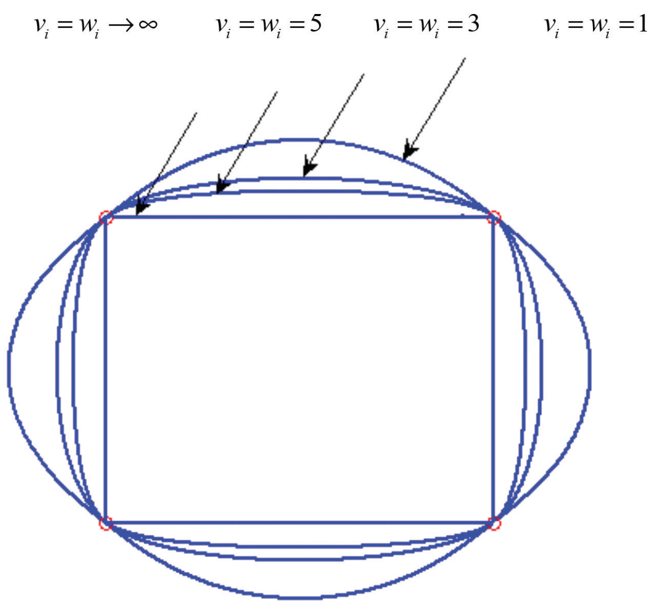


FIGURE 1. Demonstration of rational cubic function (3.1) for case 1

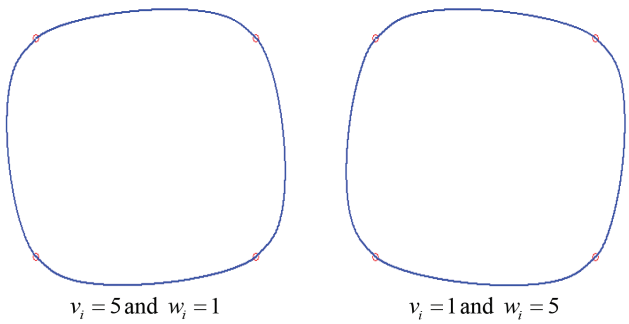


FIGURE 2. Demonstration of rational cubic function (1) for case 2

It can be observed that t_i is in normalized form and varies from 0 to 1.

ESTIMATION OF TANGENT VECTORS

A distance based choice [Sarfraz 2004] of tangent vectors D_t's at F_i´s is defined as:

For open curves:

$$\left.\begin{aligned} D_0 &= 2(P_1 - P_0) - \frac{(P_2 - P_0)}{2} \\ D_i &= a_i(P_i - P_{i-1}) + (1-a)(P_{i+1} - P_i), i = 1,2,\ldots,n-1 \\ D_n &= 2(P_n - P_{n-1}) - \frac{(P_n - P_{n-2})}{2} \end{aligned}\right\},$$

For close curves:

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

where

$$a_i = \frac{|F_{i+1} - F_i|}{|F_{i+1} - F_i| + |F_i - F_{i-1}|}, i = 0,1,\ldots,n.$$

GENETIC ALGORITHM

Genetic Algorithms (GAs) are search techniques based on the concept of evolution. In simple words, every solution, in a given well-defined search space, is represented by a bit string, called a chromosome. A Genetic Algorithm (Goldberg 1989) is applied with its three genetic search operations (selection, crossover and mutation) to create a population of chromosomes with the purpose of improving 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). In GA, a cost function generates an output from a set of input variables (a chromosome). The cost function may be a mathematical function, an experiment or a game. The objective is to modify the output in some desirable fashion by finding the appropriate values for the input variables.

The GA begins by defining a chromosome or an array of variable values to be optimized; variable values are represented in binary so the binary GA works with bits. The GA works with the binary encodings, but the cost function often requires continuous variables. Whenever the cost function is evaluated, the chromosome must be decoded. An example of a binary encoded chromosome is shown in Figure 3.

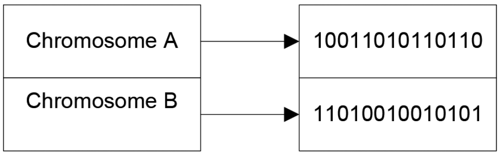


FIGURE 3. Example of binary encoding

The GA starts with a group of chromosomes known as the population. Next, the variables are passed to the cost function for evaluation. Natural selection process leads to survival of the fittest i.e. discarding the chromosomes with the highest cost. Natural selection occurs at each generation or iteration of the algorithm. Deciding how many chromosomes to keep is somewhat arbitrary. Letting only a few chromosomes survive to the next generation limits the available genes in the offspring. Keeping too many chromosomes allows bad performers a chance to contribute their traits to the next generation. We often keep 50% in the natural selection process.

Another approach to natural selection is called thresholding. In this approach all chromosomes that have a cost lower than some threshold survive. The threshold must allow some chromosomes to continue in order to have parents to produce offspring. Otherwise, a whole new population must be generated to find some chromosomes that pass the test. At first, only a few chromosomes may survive. In later generations, however, most of the chromosomes will survive unless the threshold is changed. An attractive feature of this technique is that the population does not have to be sorted.

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 and random pairing. The next step after selecting parents is mating to create one or more offspring.

Commonly used form of mating is called crossover operator which deals with two parents that produce two offspring. A crossover point is randomly selected between the first and last bits of the parents' chromosomes. First parent passes its binary code to the left of that crossover point to first offspring. Similarly second parent passes its binary code to the left of the same crossover point to second offspring. Further, 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 4.

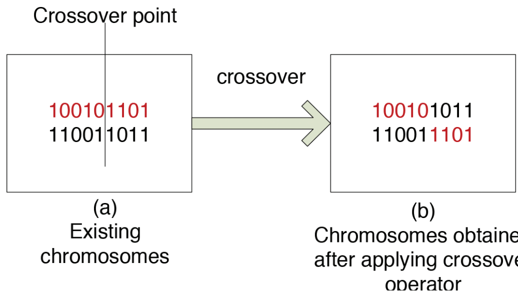


FIGURE 4. Example of crossover operator

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 5.

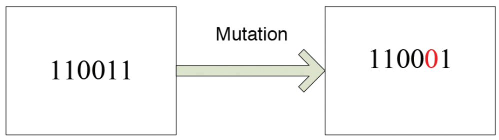


FIGURE 5. Example of mutation operator

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 6.

OPTIMAL RATIONAL CUBIC FUNCTION

In this section process of finding an optimal rational cubic function with the help of GA (Goldberg 1989) is discussed.

Suppose for $i = 1,\ldots,n$, $P_{i,j} = (x_{i,j}, y_{i,j})$, $j = 1,2,\ldots,m_i$ be the given data points, then the squared sums S_i´s of distance between $P_{i,j}$´s and their corresponding parametric points $P(t_j)$´s on the curve are determined as $S_i = \sum_{j=1}^{m_i}\left[P_i\left(u_{i,j}\right) - P_{i,j}\right]^2, i = 1,\ldots,n$ where u's are parameterized in reference to chord length parameterization explained in Section 3.1.

Case 1: When $w_i = v_i$, then rational cubic (1) would have only one shape parameter say v_i. So to find optimal curve to given data, such values of parameter v_i, are required so that the sums S_i´s are minimal. Genetic Algorithm is used to optimize this value for the fitted curve. The process will start with initial population of values of v_i chosen randomly. Successive application of search operations like selection,

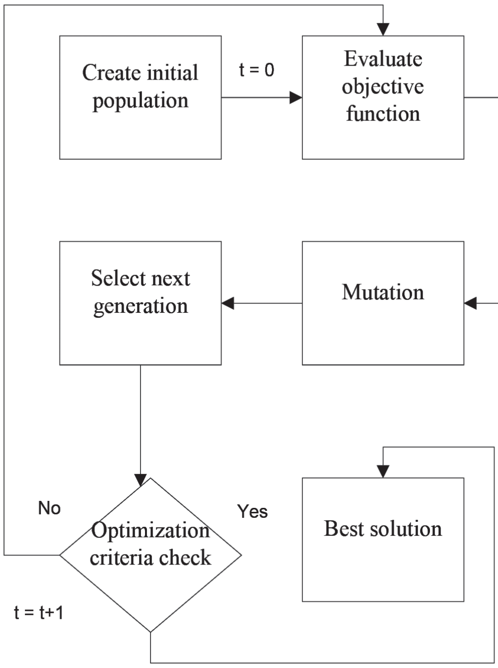


FIGURE 6. Flow diagram of Genetic Algorithm

TABLE 1. Details of digital contours and corner points

Image	Name	# of contours	# of contour points	# of initial corner points
	Fork	1	673	15
	Plane	3	915+36+54	28
	Fish	1	975	31

crossover and mutation to this population leads to optimal values of v_i.

Case 2: When $w_i \neq v_i$, then in the description of rational cubic (1) there would be two parameters w_i and v_i to be optimized so that the sums S_i´*s* attain their minimum values. For this purpose GA is applied.

INITIALIZATION

Once we have the bitmap image of a generic shape, the boundary of the image can be extracted using the method described in Section 2. After the boundary points of the image are found, the next step is to detect corner points to divide the boundary of the image into *n* segments as explained in Section 2. Each of these segments is then approximated by interpolating function described in Section 3.

CURVE FITTING

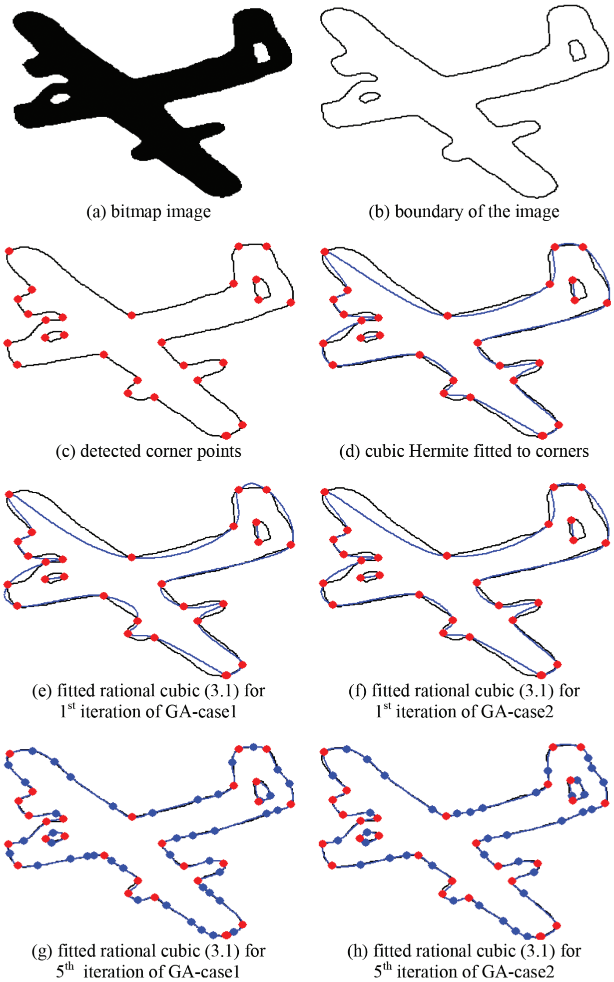
After an initial approximation for the segment is obtained, Genetic Algorithm helps to obtain better approximations to achieve optimal solution. The tangent vectors at knots are estimated by the method described in Section 3.3.

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 7(g), 7(h), 8(g), 8(h), (9(g) and 9(h). Table 2 gives details of number of intermediate points achieved during different iteration of Genetic Algorithm applied in process of curve fitting in case 1 and case 2.

TABLE 2. Number of corner points together with number of intermediate points for iterations 1, 2 and 3 of GA

Image Name	# of initial corner points	# of intermediate points in cubic interpolation with threshold value 3							
		Itr.1		Itr.2		Itr.3		Final itr.	
		Case1	Case2	Case1	Case2	Case1	Case2	Case1	Case2
Fork	15	0	0	10	9	17	17	28	25
Plane	28	0	0	21	19	31	28	39	37
Fish	31	0	0	18	18	31	30	35	38



(a) bitmap image (b) boundary of the image

(c) detected corner points (d) cubic Hermite fitted to corners

(e) fitted rational cubic (3.1) for 1st iteration of GA-case1 (f) fitted rational cubic (3.1) for 1st iteration of GA-case2

(g) fitted rational cubic (3.1) for 5th iteration of GA-case1 (h) fitted rational cubic (3.1) for 5th iteration of GA-case2

FIGURE 7. Demonstration of rational cubic function (3.1) using GA for case1 and case 2

Algorithm

Step 1: Input the data points

Step 2: Subdivide the data, by detecting corner points using the method mentioned in Section 2

Step 3: Compute the derivative values at the corner points by using formula given in Section 3.3

Step 4: Fit the rational cubic function of Section 3, to the corner points found in Step 2

Step 5: If the curve, achieved in Step 4, is optimal then go to Step 7, else find the appropriate break/intermediate points (points with highest deviation) in the undesired curve pieces. Compute the best optimal values of the shape parameters v_i and w_i. Fit rational cubic function in Section 3 to these intermediate points

Step 6: If the curve, achieved in Step 5, is optimal then go to Step 7, else add more intermediate points (points with highest deviation) and go to Step 3

Step 7: Stop

DEMONSTRATION

Curve fitting scheme proposed in Section 4 has been implemented on different images. Figure 11(a) represents the original image, (b) shows outline of the image, (c) demonstrates corner points, (d) presents fitted Hermite curve to the corners along with boundary of the image (e) and (f) give fitted outline for 1st iteration using Genetic Algorithms together with corner points and boundary for case 1 and case 2 of rational cubic function 3.1, respectively, and (g) and (h) depict fitted outline for 5th iteration using Genetic Algorithms together with corner points breakpoints and boundary for case 1 and case 2 of rational cubic function 3.1, respectively.

Figures 8 and 9 can also be described in similar fashion. Figures 10, 11 and 12 present comparison of both the cases discussed in Section 3 of rational cubic function 3.1, while applying proposed scheme. In these figures solid line represents minimum cost for case 1 in different iterations of GA, whereas dashed line shows the minimum cost for different iterations in case2. From these figures it can be noticed that case 2 gives better results than case 1 as far as minimum cost is concerned. Moreover it can be seen in Table 2 that number of breakpoints in case 2 is less than as in case 1 for different iterations of GA in different images. So it may be concluded that proposed scheme gives slightly better results in case when it is applied using rational cubic with two parameters.

Figures 13-17 show behaviors of best, worst and mean values of fitness function for the image of fish on running GA again and again. It can be observed in Figures 13 and 15 that the best, worst and mean values of cost function coincide after iteration 6, whereas Figure 11 depicts the case where they never become equal. While Figure 16 shows that the functions start decreasing initially and become equal at 7th iteration but after that

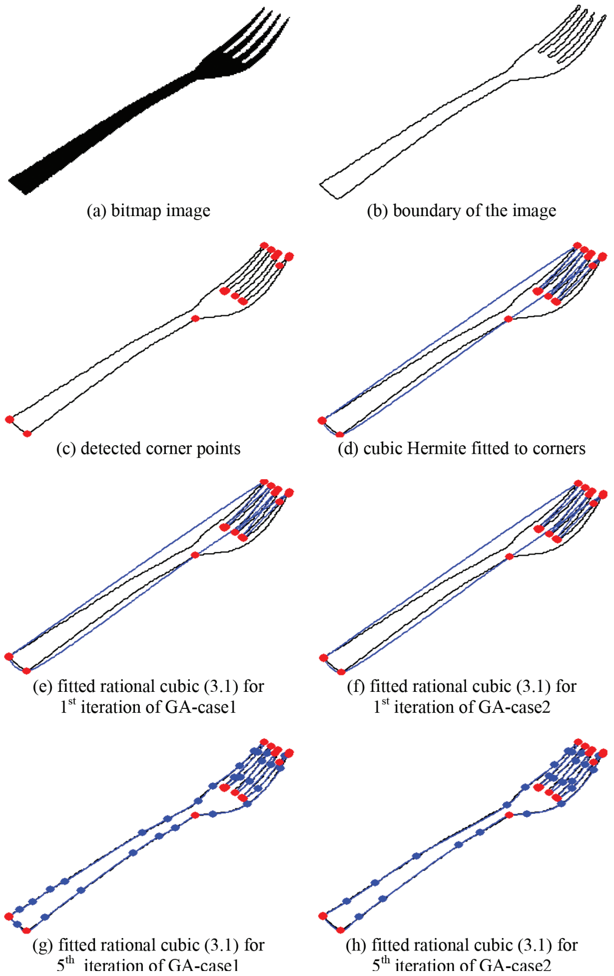


FIGURE 8. Demonstration of rational cubic function (3.1) using GA for case1 and case 2

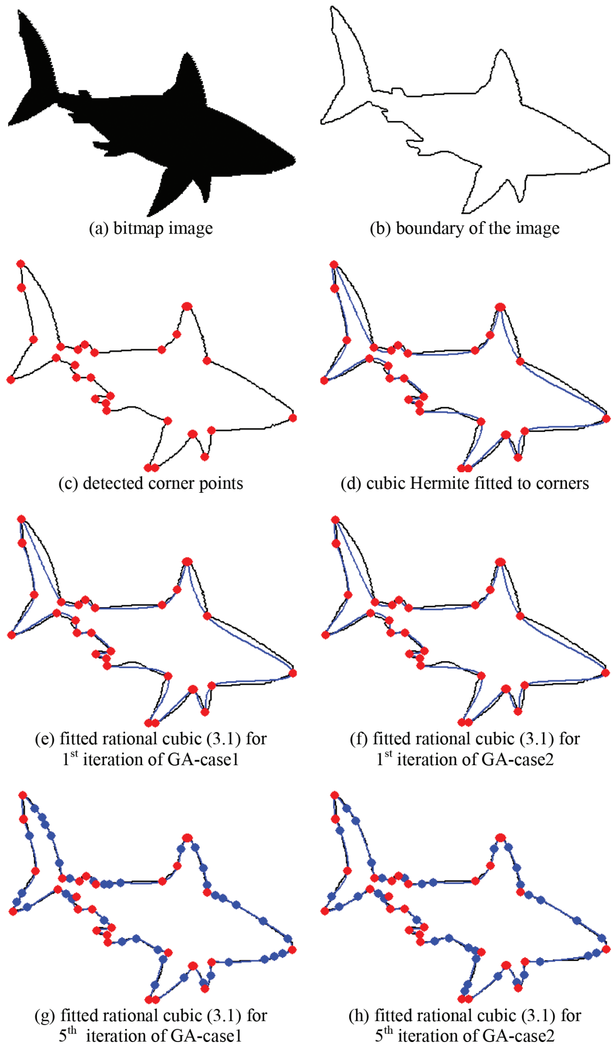


FIGURE 9. Demonstration of rational cubic function (3.1) using GA for case1 and case 2

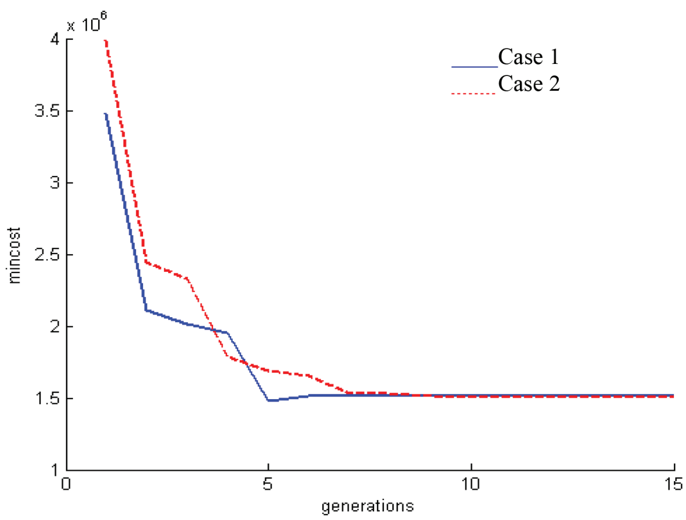


FIGURE 10. Graph of minimum cost for 'plane' at different generation for case 1 and case 2

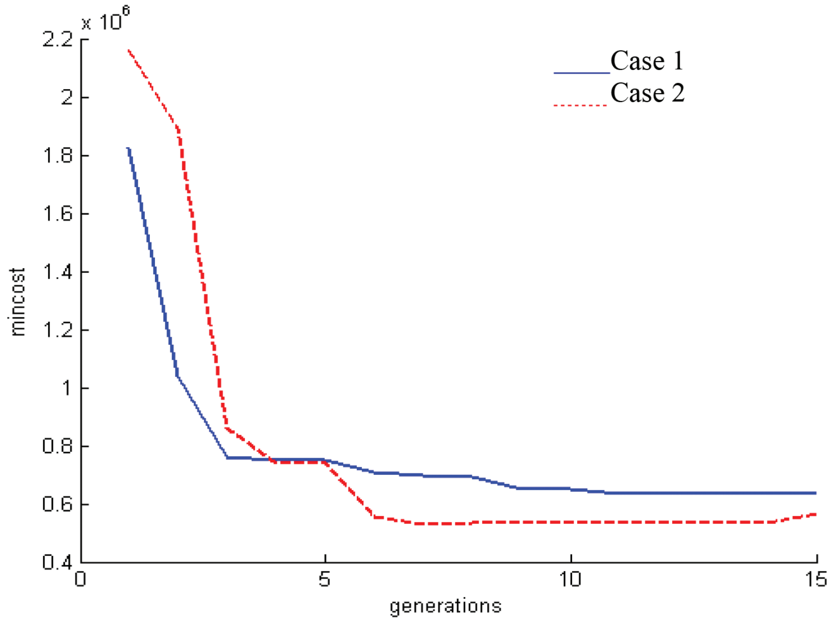


FIGURE 11. Graph of minimum cost for 'fork' at different generation for case 1 and case 2

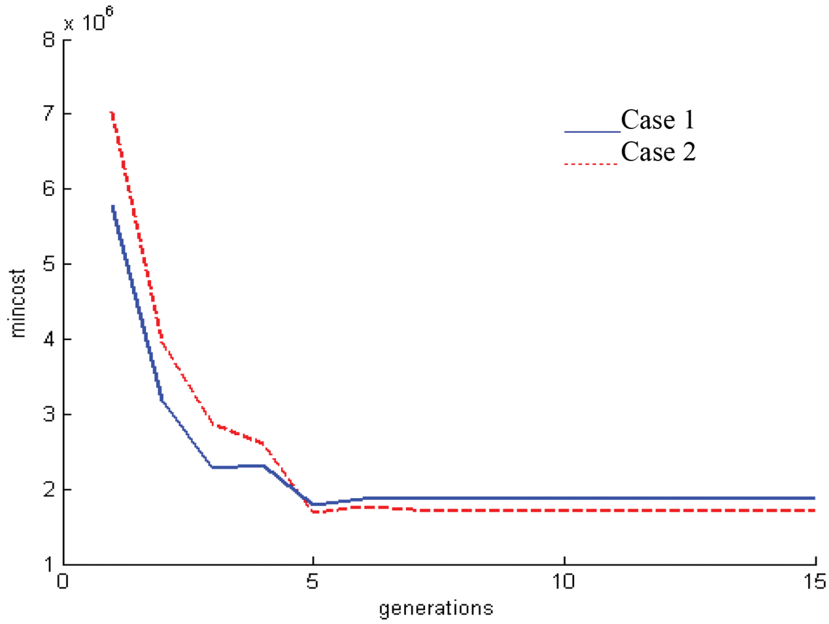


FIGURE 12. Graph of minimum cost for 'fish' at different generation for case 1 and case 2

all the functions increase in 8^{th} iteration and then seem to be constant onwards. However Figure 17 presents that all the functions coincide very early like in 2^{nd} iteration.

Figures 16-20 give the percentage of stopping criteria met by GA for the images of plane, fork and fish, respectively and the parameters used while applying GA are given in Table 3.

CONCLUSION

A scheme for reverse engineering of planar objects is presented which vectorizes the generic shapes. A rational cubic function with one shape parameter and two shape parameters is used for curve fitting and a soft computing technique genetic algorithm is applied to find optimal values of the parameters in the description of the rational cubic function. The method proposed starts with initial

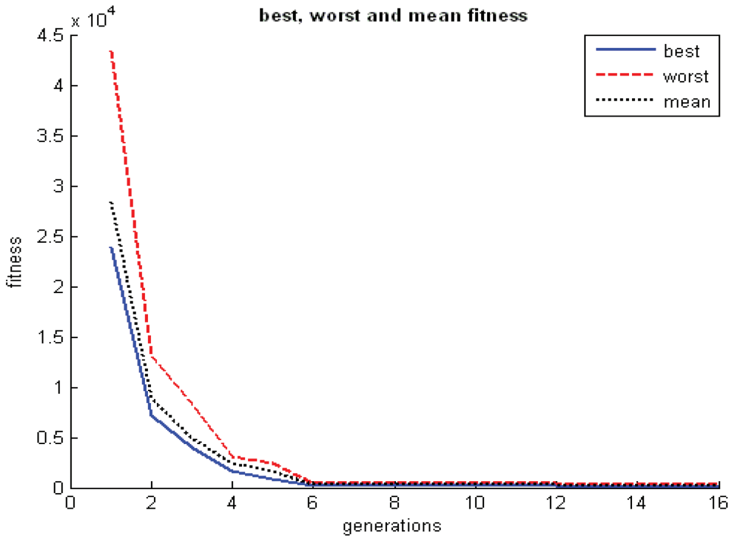


FIGURE 13. Graph of best, worst and mean fitness function

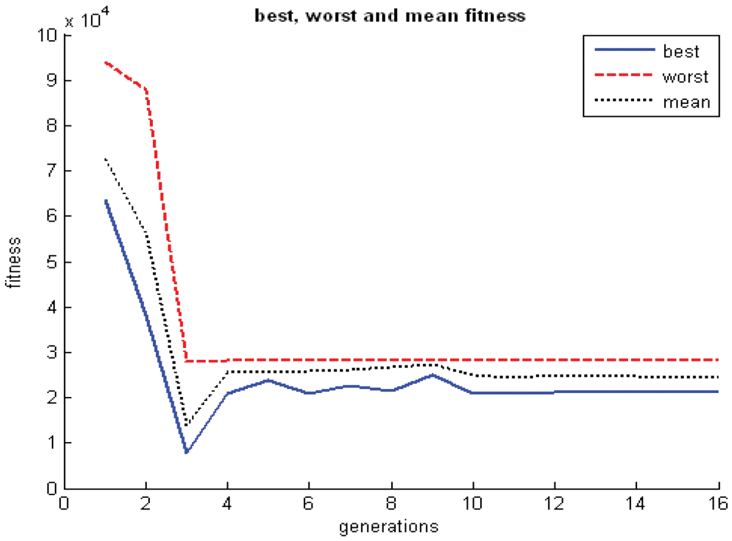


FIGURE 14. Mix behaviour of best, worst and mean fitness functions

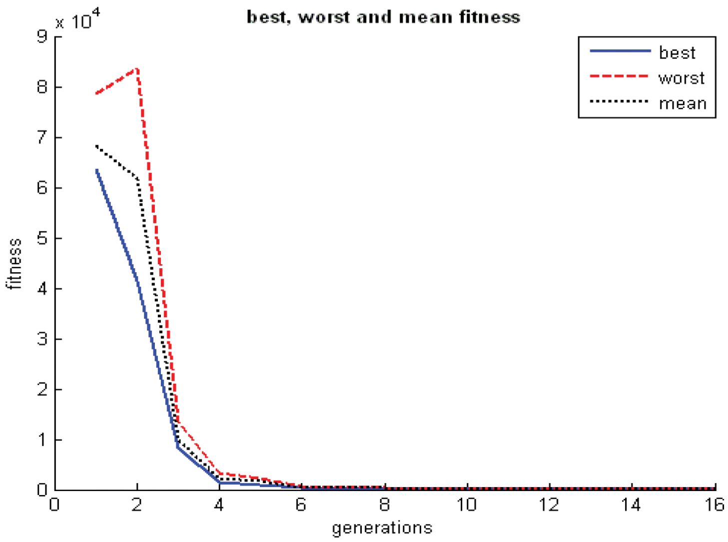


FIGURE 15. Another graph of best, worst and mean fitness functions

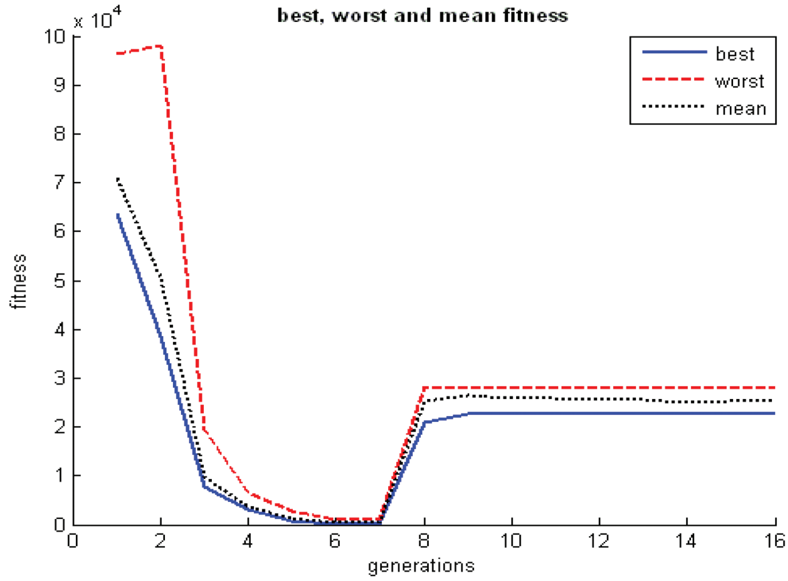


FIGURE 16. Divergent behaviour of best, worst and mean finess functions

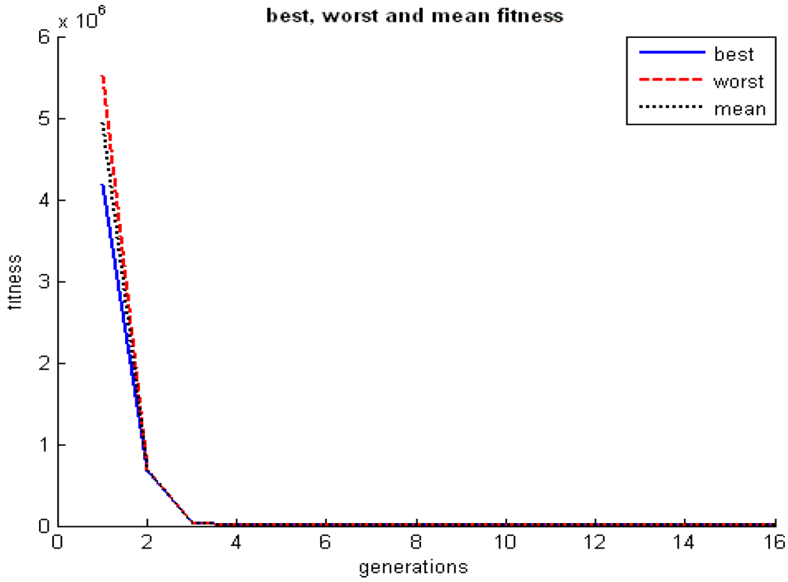


FIGURE 17. Early convergence of fitness functions

TABLE 3. Parameters of GA

Sr. No.	Name	Values
1	Population size	25
2	Genome length	15
3	Selection rate	0.5
4	Mutation rate	0.01

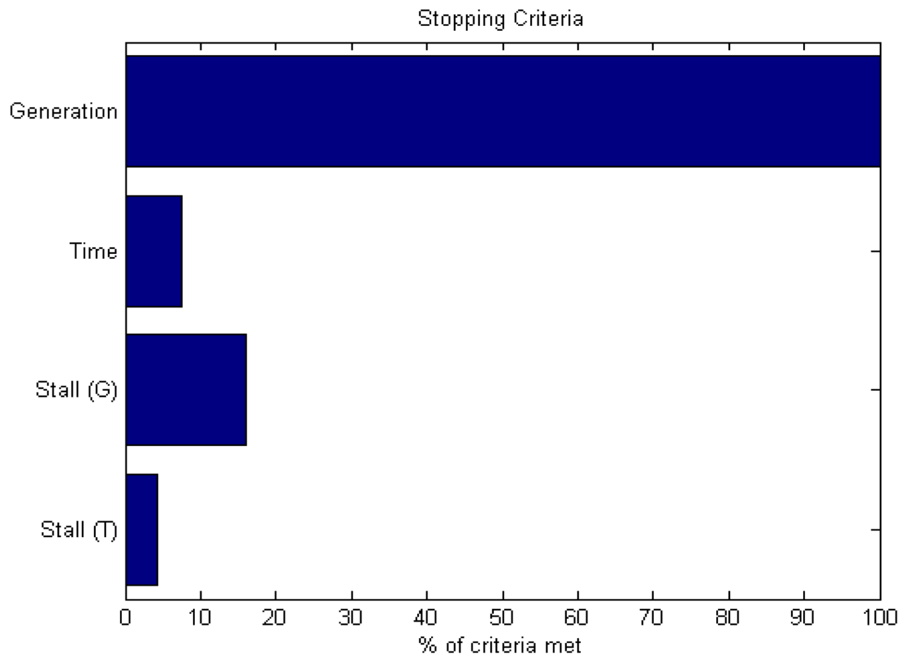


FIGURE 18. Stopping crieteria met by GA for image of plane

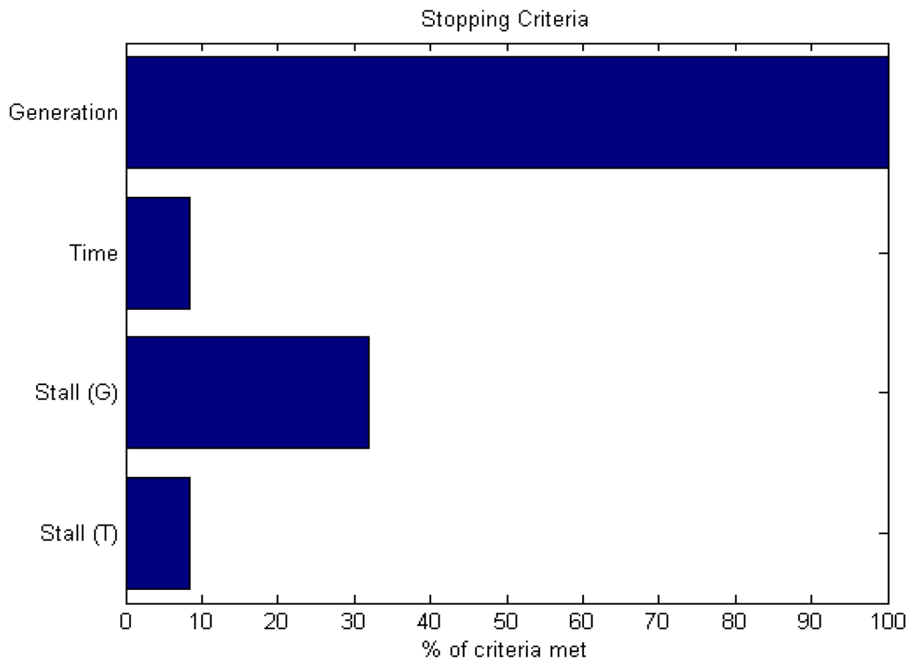


FIGURE 19. Stopping crieteria met by GA for image of fork

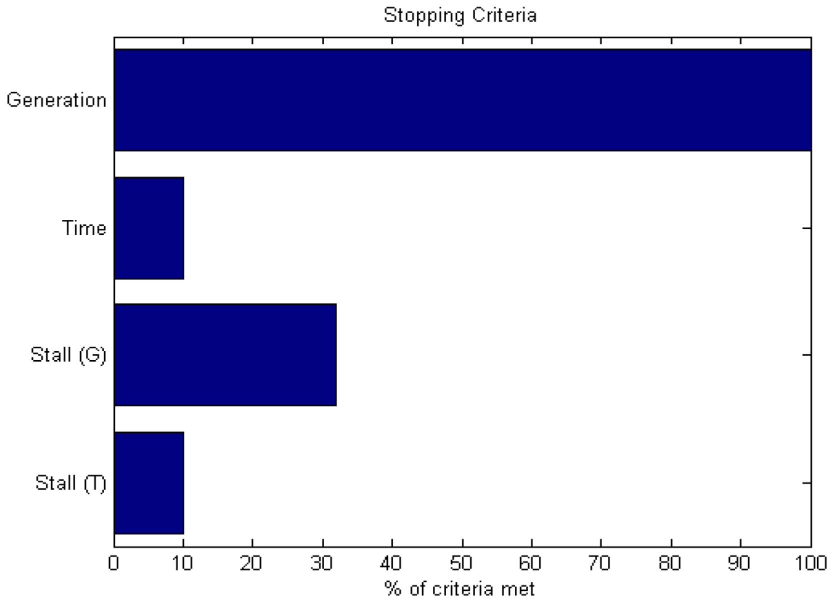


FIGURE 20. Stopping crieteria met by GA for image of fish

random population of parameters and find those values of the parameters which can assure best optimal curve to the data extracted by bitmap images. A comparison between both the cases of rational cubic is also done which proves that the rational cubic with two parameters gives slightly better results as far as minimum error and breakpoints are concerned. The scheme presented is fully automatic and it also ensures computational efficiency as far as curve fitting is concerned. This work might be extended to the 3D models in future.

ACKNOWLEDGEMENT

The authors highly acknowledge the referee's valuabale comments on the improvement of the manuscript.

REFERENCES

Avrahami, G. & Pratt, V. 1991. Sub-pixel edge detection in character digitization. *Raster Imaging and Digital Typography II* : 54-64.

Beus, H.L. & Tiu, S.S.H. 1987. An improved corner detection algorithm based on chain coded plane curves. *Pattern Recognition* 20: 291-296.

Cabrelli, A.C. & Molter, U.M. 1990. Automatic representation of binary images. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 12(12): 1190-1196.

Chandrasekaran, M., Muralidhar, M., Murali Krishna, C. & Dixit, U.S. 2010. Application of soft computing techniques in machining performance prediction and optimization: A literature review. *The International Journal of Advanced Manufacturing Technology* 46: 445-464.

Chetrikov, D. & Zsabo, S. 1999. A simple and efficient algorithm for detection of high curvature points in planar curves. *Proceedings of the 23rd Workshop of Austrian Pattern Recognition Group*.

Davis, L. 1979. Shape matching using relaxation techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1(1): 60-72.

Freeman, H. 1961. On the encoding of arbitrary geometric configurations. *IEEE Transactions on Electronic Computer* 10(2): 260-268.

Goldberg, D.E. 1989. *Genetic Algorithms in Search Optimization and Machine Learning*. Reading, MA: Addison Wesley.

Hölzle, G.E. 1983. Knot placement for piecewise polynomial approximation of curves. *Computer Aided Design* 15(5): 295-296.

Hou, Z.J. & Wei, G.W. 2002. A new approach to edge detection. *Pattern Recognition* 35(7): 1559-1570.

Itoh, K. & Ohno, Y. 1993. A curve fitting algorithm for character fonts. *Electronic Publishing* 6(3): 195-198.

Kirkpatrick, S., Gelatt Jr., C.D. & Vecchi, M.P. 1983. Optimization by simulated annealing. *Science* 220(4598): 671-680.

Plass, M. & Stone, M. 1983. Curve-fitting with piecewise parametric cubics. *Computer Graphics* 17(3): 229-239.

Sarfraz, M. & Khan, M.A. 2004. An automatic algorithm for approximating boundary of bitmap characters. *Future Generation Computing Systems* 20(8): 1327-1336.

Sarfraz, M. 2004. Some algorithms for curve design and automatic outline capturing of images. *International Journal of Image and Graphics* 4(2): 301-324.

Sarfraz, M. & Rasheed, A. 2007. 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, Korea*, March 11-15.

Sarfraz, M. & Razzak, M.F.A. 2002. An algorithm for automatic capturing of font outlines. *Computers and Graphics* 26(5): 795-804.

Sarfraz, M. & Khan, M. 2000. Towards automation of capturing outlines of Arabic fonts. *Proceedings of the third KFUPM workshop on information and computer science: software development for the new millennium, Saudi Arabia, October 22–23*.

Sarfraz, M. & Khan, M.A. 2002. Automatic outline capture of Arabic fonts. *Information Sciences* 140(3-4): 269-281.

Sarfraz, M., Asim, M.R. & Masood, A. 2004. Capturing outlines using cubic Be´ zier curves. *Proceedings of 1st IEEE International Conference on Information and Communication Technologies: From Theory to Application, April 19-23*. pp. 539-540.

Sarfraz, M. & Raza, S.A. 2001. Capturing outline of fonts using genetic algorithm and splines. *The Proceedings of IEEE International Conference on Information Visualization-IV-UK, July 25-27*.

Sarfraz, M., Riyazuddin, M. & Baig, M.H. 2005. Capturing planar shapes by approximating their outlines. *Journal of Computational and Applied Mathematics* 189: 494-512.

Sarfraz, M. & Raza, A.A. 2002. Visualization of data with spline fitting: A tool with a genetic approach. *Proceedings of International Conference on Imaging Science, Systems, and Technology, Las Vegas, Nevada, USA, June 24-27*.

Sarfraz, M., Masood, A. & Asim, M.R. 2006. A new approach to corner detection. *Computer Vision and Graphics* 32: 528-533.

Sarfraz, M., Irshad, M. & Hussain, M.Z. 2012, Vectorization of Image Outlines using Rational Spline and Genetic Algorithm, *The Proceedings of The International Conference on Image, Vision and Computing (ICIVC 2012)*. Shanghai, China, August 25-26, 2012, IASCIT Press Singapore.

Schneider, P.J. 1990. An algorithm for automatically fitting digitized curves. *Graphics Gems*. pp. 612-626.

Sohel, F.A., Karmakar, G.C., Dooley, L.S., Arkinstall, J. 2005. Enhanced Be´ zier curve models incorporating local information. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, USA, March 18-23*. .

Tang, Y.Y., Yang, F. & Liu, J. 2001. Basic processes of chinese character based on cubic B-spline wavelet transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(12): 1443-1448.

Teh, C.H. & Chin, R.T. 1989. On the detection of dominant points on digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11(8): 859-873.

Muhammad Sarfraz*
Department of Information Science
Kuwait University
Kuwait

Misbah Irshad & Malik Zawwar Hussain
Department of Mathematics
University of the Punjab, Lahore
Pakistan

*Corresponding author; email: prof.m.sarfraz@gmail.com

Received: 15 June 2012
Accepted: 25 January 2013