Optimal Curve Fitting to Digital Data

Muhammad Sarfraz
King Fahad University of Petrolium and Minerals
Department of Information and Computer Science
KFUPM # 1510, Dhahran, Saudi Arabia
sarfraz@kfupm.edu.sa

ABSTRACT

An optimal curve fitting technique has been developed which is meant to automatically provide a fit to any ordered digital data in plane. A more flexible class of rational cubic functions is the basis of this technique. This class of functions involves two control parameters, which help to produce optimal curve fit. The curve technique has used various ideas for curve design. These ideas include end-point interpolation, intermediate point interpolation, detection of characteristic points, and parameterization. The final shape is achieved by stitching the generalized Bézier cubic pieces with an ideally acceptable smoothness.

Keywords

Data, Curve fitting, Characteristic points, Interpolation, Spline.

1. INTRODUCTION

Designing of curves, especially those curves which are robust and easy to control and compute, has been one of the significant problems of Computer Graphics. Specific applications including Font Designing [Hus89], Capturing Hand-Drawn Images [Sar01a, Sar02b] on computer screens, Data Visualization [Bro93, Gre86a, Sar02c, Sar02d] and Computer-Supported Cartooning are main motivations towards curve designing. In addition, various other applications [Far94, Gre90b] in CAD/CAM/CAGD are also a good reason to study this topic.

In curve designing, the class of rational cubic functions [Far94, Gre90b], is one of the most powerful tools. They can define space curves and curves with inflections. This paper presents a simple and effective method for optimal curve design for an economical smooth curve fit to a large digital data.

The methodology, in this paper, differs to the traditional approaches. It is multipurpose and can solve problems, of various natures, in minimal cost and under the same umbrella. It also demonstrates the output in a visually pleasant way as the spline model is more than C^1 smooth. It also differs in the sense that it is based on appropriately computed parameters in the description of the rational spline model. A major difference lies with the subdivision methodology to conquer the desired solution. Another major difference lies in the curve model for the description of design curve. The outline capturing technique, instead of traditional Bézier cubics, is based upon a rational cubic model.

The subdivision process, in this paper, is managed as follows. In the case of large data points, the significant points are located. These significant points are identified by manipulating high curvature points using the algorithm in [Che99] (however, modifying the algorithm in [Dav79] could also be proposed as an alternate solution.) Then, if needed, it modifies and enhances the list of significant points by introducing intermediate points. This idea of intermediate points, again, assists in the further subdivision. The collection of all these points will be called as *characteristic points* through out the paper. The characteristic points play important role in the overall display of the final shape.

The rational cubic is converted to rational cubic Hermite formulation for the manipulation of the curve segment, which interpolates the two endpoints of the curve segment and the consecutive pieces of curve segments have equal tangents at each characteristic point. Accordingly, the design curve will be C^1 continuous. In addition, the proposed curve model has the property to interpolate an

Journal of WSCG, Vol.11, No.1., ISSN 1213-6972
WSCG'2003, February 3-7, 2003, Plzen, Czech Republic.

intended intermediate point, which can be of great help for the optimal fit.

The organization of the paper is made as follows. Section 2 explains briefly the rational cubic segment and its various properties including the intermediate point manipulation. The piecewise design curve method is explained in Section 3 together with the choices of distance-based and spline-based tangent approximations at characteristic points. The issue of subdivision, by detecting the characteristic points (significant points and intermediate points), is discussed in Section 4. The issue of detecting the characteristic points has also been explained in this section. Section 5 summarizes the whole discussion in the form of an algorithm. This algorithm is applicable to visualize large data with minimum number of data utilization. Section 6 is meant for the demonstration, two examples have been explained for the outline capture of font images. Section 7 concludes the paper.

2. RATIONAL CUBIC INTERPOLANT

Let P_0, P_1 be the end points and T_0, T_1 be the end tangents of a curve segment, then the rational cubic Hermite function manipulating the curve segment is

$$P(t) = \frac{P_0(1-t)^3 + vP_{0,1}t(1-t)^2 + wP_{1,1}t^2(1-t) + P_1t^3}{(1-t)^3 + vt(1-t)^2 + wt^2(1-t) + t^3}, \tag{1}$$

where

$$\left.\begin{aligned} P_{0,1} &= P_0 + T_0/v \\ P_{1,1} &= P_3 - T_1/w \end{aligned}\right\} \tag{2}$$

and

$$v, w \geq 0 \tag{3}$$

From (1), it can be verified that

$$P(0) = P_0,\ P(1) = P_1,$$
$$P'(0) = T_0,\ P'(1) = T_1.$$

Thus the curve segment obtained by (1) is C^1. Equation (2) represents the tangents in terms of the given control points and the shape parameters. Equation (1) can be written as

$$P(t) = R_0(t)P_0 + R_1(t)P_{0,1} + R_2(t)P_{1,1} + R_3(t)P_1, \tag{4}$$

where function R_i, $i = 0,1,2,3$ are Bernstein Bézier like basis functions such that

$$\sum_{i=0}^{3} R_i(t) = 1. \tag{5}$$

It shows that the curve segment will always lie in the convex hull of the control points.

2.1. Some Observations

We observe the following properties of the interpolant defined by equation (1):

(a) For $v = w = 3$, the rational cubic vanishes and we are left with a cubic (Bézier Cubic).

(b) The curve always passes through P_0 and P_1.

(c) If $v \to \infty$, then the curve exhibits the *biased tension behavior to the left* and is pulled towards the control point P_0.

(d) If $w \to \infty$, then the curve exhibits the *biased tension behavior to the right* is pulled towards the control point P_1.

(e) If $v, w \to \infty$, then the curve exhibits the interval tension behavior and approaches to the linear interpolant.

(f) If $v, w \to 0$, then the curve gets loosened. The curve bulges outside the convex hull for negative values. Thus, a relatively looser curve, as compared to Bézier cubic curve, is also achievable.

2.2. Intermediate Point Interpolation

Definition 1. A point S, on a curve $P(t)$, $t \in (0,1)$, will be called an intermediate point if $P(t) = S$ for $t \in (0,1)$.

In certain applications, where an optimally good fit is needed, it is desired to subdivide some of the curve pieces at appropriate intermediate points. Interpolation of an intermediate point may be a very useful tool in controlling a curve. The parametric curve defined by (1) can be used to interpolate an intermediate point. If

$$P_0 = (x_0, y_0),\ P_{0,1} = (x_1, y_1),$$
$$P_{1,1} = (x_2, y_2),\ P_1 = (x_3, y_3),\ S = (x_s, y_s),$$

and S is an intermediate point with a parametric value u then $P(u) = S$ implies the following:

$$v = \frac{F_0(u)A + F_3(u)B}{F_2(u)E} \tag{6}$$

and

$$w = \frac{F_0(u)C + F_3(u)D}{F_1(u)E} \tag{7}$$

where

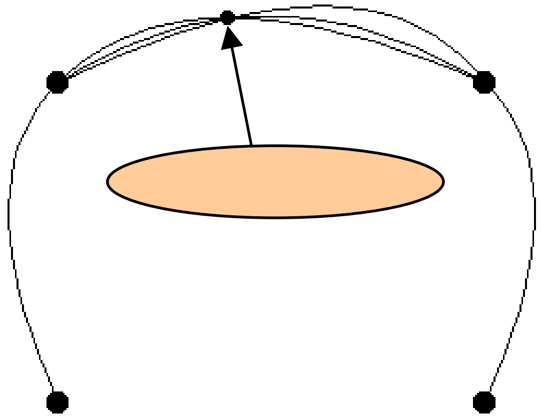


Figure 1. Determination of Intermediate Point.

$$\left.\begin{aligned} F_0(u) &= (1-u)^3 \\ F_1(u) &= u(1-u)^2 \\ F_2(u) &= u^2(1-u) \\ F_3(u) &= u^3 \end{aligned}\right\}, \qquad (8)$$

and

$$\left.\begin{aligned} A &= (x_0-x_s)(y_1-y_s)-(x_1-x_s)(y_0-y_s) \\ B &= (x_3-x_s)(y_1-y_s)-(x_1-x_s)(y_3-y_s) \\ C &= (x_2-x_s)(y_0-y_s)-(x_0-x_s)(y_2-y_s) \\ D &= (x_2-x_s)(y_3-y_s)-(x_3-x_s)(y_2-y_s) \\ E &= (x_2-x_s)(y_1-y_s)-(x_1-x_s)(y_2-y_s) \end{aligned}\right\}. \qquad (9)$$

For different values of u ranging form 0 to 1, we will have different values of *v* and *w*. Thus, we get different curves passing through S, see Figure 1. In this case u can be used as a shape parameter. If $u \to 0$, the part of the curve segment between P_0 and S will be tightened and if $u \to 1$, the part of the curve segment between S & P_1 will be tightened.

3. INTERPOLATION DESIGN CURVE

Let us generalize the idea of curve design for any given data set. Using the method explained in Section 2, we can generate the curve segments for any given number of data points. Joining these segments, we can generate the desired Design Curve. The curve thus obtained will be C^1. The procedure for curve design is as follows.

Let $F_i \in R^m, i \in Z,$ be data values given at the distinct knots $t_i \in R\ , i \in Z,$ with interval spacing $h_i := t_{i+1} - t_i > 0$. Also, let $F_i \in R^m, i \in Z,$ denote the first derivative values defined at the knots. Then the generalized form of the rational cubic, in the form a parametric C^1 piecewise piecewise rational cubic Hermite function $P: R \to R^m$, is defined by

$$P|_{(t_i,t_{i+1})}(t) := \frac{(1-\theta_i)^3 F_i + \theta_i(1-\theta_i)^2 v_i V_i + \theta_i^2(1-\theta_i) w_i W_i + \theta_i^3 F_{i+1}}{(1-\theta_i)^3 + v_i\theta_i(1-\theta_i)^2 + w_i\theta_i^2(1-\theta_i) + \theta_i^3}, \qquad (10)$$

where

$$\theta_i \equiv \theta_i(t) = \theta|_{(t_i,t_{i+1})}(t) := (t-t_i)/h_i, \qquad (11)$$

and

$$\mathrm{V_i} := \mathrm{F_i} + \frac{1}{v_i} h_i D_i, \quad \mathrm{W_i} := \mathrm{F_{i+1}} - \frac{1}{w_i} h_i D_{i+1}. \qquad (12)$$

This form is more economical for computational purposes. We have made use of a rational Bernstein-Bézier representation, where the control points $\{F_i, V_i, W_i, F_{i+1}\}$ are determined by imposing the Hermite interpolation conditions

$$P(t_i) = F_i \text{ and } P^{(1)}(t_i) = D_i,\ i \in Z \qquad (13)$$

In most of the applications, the tangent information are not provided. We define two choices for tangent vectors D_i at F_i as follows:

3.1. A Distance-Based Choice

For open curve:

$$\left.\begin{aligned} D_0 &= 2(F_1-F_0)-F_2-F_0)/2, \\ D_i &= a_i(F_i-F_{i-1})+(1-a_i)(F_{i+1}-F_i), \\ &\qquad\qquad i = 1,..,n-1, \\ D_n &= 2(F_n-F_{n-1})-F_n-F_{n-2})/2. \end{aligned}\right\} \qquad (14)$$

For Close Curve:

$$\left.\begin{aligned} &F_{-1} = F_{n-1,} F_{n+1} = F_1, \\ &D_i = a_i(F_i-F_{i-1})+(1-a_i)(F_{i+1}-F_i),\ i = 0,..,n, \end{aligned}\right\} \qquad (15)$$

where

$$a_i = \frac{|F_{i+1}-F_i|}{|F_{i+1}-F_i| + |F_i-F_{i-1}|},\ i = 0,..,n. \qquad (16)$$

This choice of tangents provides nice and pleasing results.

3.2. A Spline-Based Choice

Since, the default case of an underlying function is the following cubic:

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

where

$$\mathrm{V_i} := \mathrm{F_i} + \frac{1}{3} h_i D_i, \quad \mathrm{W_i} := \mathrm{F_{i+1}} - \frac{1}{3} h_i D_{i+1}.$$

This is a C^1 Hermite cubic. For smoother splinning, we impose C^2 constraints across the knots, which leads to the following systems of equations:

$$c_{k-1} D_{k-1} + 2(c_{k-1} + c_k) D_k + c_k D_{k+1} = b_k (F_{k+1} - F_k) + b_{k-1}(F_k - F_{k-1}) \quad (1)$$

for $k = 2, ..., n-1$, where $c_i = 1/h_i$, $b_i = 3c_i / h_i$. The above system of equations provides (n-2) equations for n unknowns, $D_1,\ldots,D_n$, and the additional equations come from the given end conditions. The equations for Type I natural end conditions they are

$$2c_1 D_1 + c_1 D_2 = b_1 (y_2 - y_1)$$

and

$$c_{n-1} m_{n-1} + 2c_{n-1} D_n = b_{n-1}(F_n - F_{n-1}).$$

For Type II periodic end conditions, they are

$$2(c_1 + c_{n-1}) D_1 + c_1 D_2 + c_{n-1} D_{n-1} = b_1 (F_2 - F_1) + b_{n-1}(F_n - F_{n-1})$$

and $D_1 = D_n$. The linear system of equations that occurs when Type 1 or 2 end conditions are used is tridiagonal and diagonally dominant, and hence has a unique solution for D_i's. As far as the computation method is concerned, it is much more economical to adopt the LU-decomposition method to solve the tridiagonal system. Therefore, the above discussion can be concluded in the following:

Theorem 1 *For the constraints in Section 3.2, the spline solution of the Rational spline exists and is unique.*

3.3.Some Remarks

Here are some important remarks to be noted.

3.3.1. Remark 1

A user has a choice to pick and chose any of the above mentioned two choices for rational spline manipulation.. For the implementation purposes, in this paper, the author would prefer to chose the second choice of Subsection 3.2 as it provides a relatively smoother output. However, there is a freedom to chose the option of Subsection 3.1 too. This option also provides reasonably good results.

3.3.2. Remark 2

It should be noted that, in case of first option of tangents, the curve scheme would be exactly C^1. But, the second option will raise the smoothness to a better level as the tangents are computed through C^2 cubic spline constrains.

4. OPTIMAL DESIGN CURVE

When the data is too large, the interpolating scheme in Section 3 may not be advisable to use as it costs too much in terms of computations. For example, for a set of 100001 points, 10000 pieces are needed to be stitched. Specifically, when the data points are closed to each other, there is no need for so many segments to be manipulated. Therefore, some other approach is needed to be discussed so that an optimal and economical design curve is achieved. The proposed method of curve fit, in this section, consists of all the following steps.

(a) The given data is subdivided at Characteristic points. These Characteristic points are determined from the given data points (the detail about the characteristic points are given in the following Subsection 4.1)

(b) The spline method, in Section 3, is used for the piecewise fit to the data points between two consecutive characteristic points.

(c) If the design curve still needs improvement, the parameters, in the description of the rational function, can be utilized for the best fit. These parameters are evaluated, using intermediate point interpolation method in Section 2.2, in such a way that the corresponding pieces of curves are optimal to the given data.

(d) If the curve fit in (c) is not as optimal as desired, further subdivision process is made at appropriate data points; this will enhance the list of characteristic points (this subdivision process is explained in Subsection 4.2.)

4.1. Detection of Characteristic Points

The characteristic points are those points, which partition the data into various pieces. An approach based on curvature analysis, with numerical technique, has been proposed. The detection of the characteristic points actually also depends on how closed the data points are from each other. The details of this theory are provided in the following Subsubsections.

4.1.1. Characteristic Points as Corner Points

In the first step, the characteristic points are searched on the basis of computation of high curvatures at each data point. The strategy of this procedure is based on the method of [Che99].

The proposed two-pass algorithm defines a corner in a simple and intuitively appealing way, as a location where a triangle of specified size and opening angle can be inscribed in a curve.

A curve is represented by a sequence of points $\mathbf{p}_i$ in the image plane. The ordered points are densely sampled along the curve, but contrary to the other four algorithms, no regular spacing between them is assumed. A chain-coded curve can also be handled if converted to a sequence of grid points. In the first pass the algorithm scans the sequence and selects candidate corner points. The second pass is post-processing to remove superfluous candidates.

First pass. In each curve point **p** the detector tries to inscribe in the curve a variable triangle (p^-, p, p^+) constrained by a set of simple rules:

$$d_{\min}^2 \le \left|p - p^+\right|^2 \le d_{\max}^2,$$

$$d_{\min}^2 \le \left|p - p^-\right|^2 \le d_{\max}^2,$$

and

$$\alpha \le \alpha_{\max},$$

where $\left|p - p^+\right| = |a| = a$ is the distance between p and p^+, $\left|p - p^-\right| = |b| = b$ the distance between p and p^- and α is the opening angle of the triangle. The latter is computed as

$$\alpha = \cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$$

Variations of the triangle that satisfy the conditions (7) are called admissible. Search for the admissible variations starts from p outwards and stops if any of the conditions (7) is violated. Among the admissible variations, the least opening angle $\alpha(p)$ is selected. $\Pi - |\alpha(p)|$ is assigned to p as the *sharpness* of the candidate. If no admissible triangle can be inscribed, p is rejected and no sharpness is assigned.

Second pass: A corner detector can respond to the same corner in a few consecutive points. Similarly to edge detection, a post-processing step is needed to select the strongest response by discarding the non-maxima points.

A candidate point p is discarded if it has a sharper valid neighbor p_v: $\alpha(p) > \alpha(p_v)$. A candidate point p_v is a valid neighbor of p if $\left|p - p_v\right|^2 \le d_{\max}^2$.

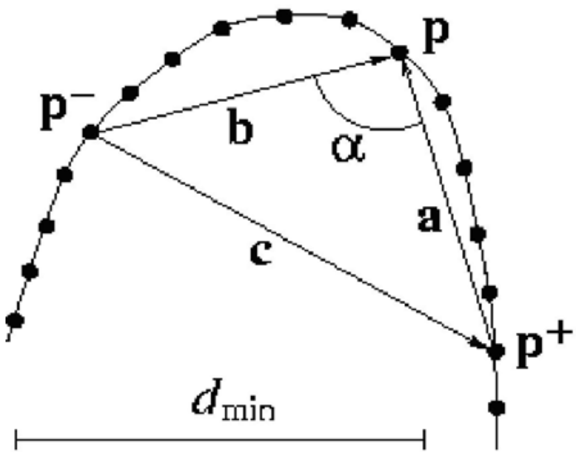


Figure 2. Corner detection, first pass.

Parameters: $d_{\min}$, d_{max} and α_{max} are the parameters of the algorithm The upper limit d_{max} is necessary to avoid false sharp triangles formed by distant points in highly varying curves. α_{max} is the angle limit that determines the minimum sharpness accepted as high curvature. In practice, we often set $d_{max} = d_{min} + 2$ and tune the remaining two parameters, d_{min} and α_{max}. The default values are $d_{min} = 7$ and $\alpha_{max} = 150^{\circ}$.

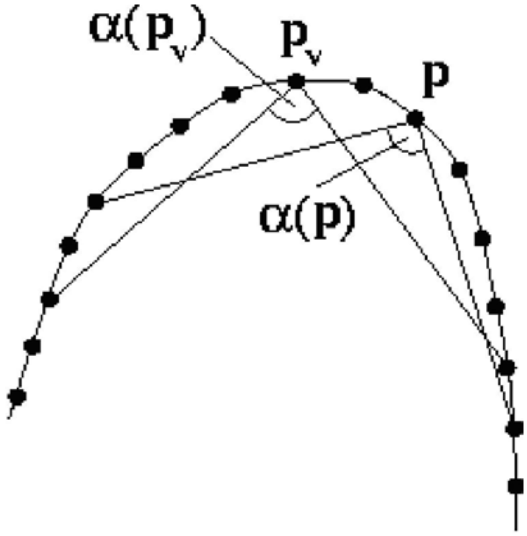


Figure 3. Corner detection, second pass.

4.1.2. More Characteristic Points

The Second step is adopted to get further characteristic points. We calculate the length of polygonal line segments between each two consecutive characteristic points (so far calculated) and the corresponding data points in between.

If these calculated lengths are greater than the defined threshold value, we break the segment into two or more segments and take intermediate point as characteristic point. This process is repeated until each segment length satisfies the threshold value criteria. The resultant set of characteristic points is finally utilized for the piecewise curve generation.

4.2. Piecewise Optimal Fit

We divide the whole set of data points into segments in such a way that first and last point of all the segments are characteristic points obtained by the methodology explained in Subsection 4.1. Each segment will be an ordered subset of the universal set of data points. A piecewise treatment is made for the curve fitting approach. All the pieces are stitched finally using the method in Section 3. For each segment of the data points, following procedure is adopted:

4.2.1. Step 1

Suppose, for $i = 0,1,2,......n-1,$ the data segments

$$G_i = \{P_{i,j} = (x_{i,j}, y_{i,j}), j = 0,1,2,.....m+1\} \quad (17)$$

are given as the ordered sets of the universal set of the data points. Then the squared distances $d_{i,j}$'s between $P_{i,j}$'s and their corresponding points $P(t_j)$'s on the curve are computed as:

$$\begin{aligned} d_{i,j} &= [P_i(u_{i,j}) - P_{i,j}]^T \cdot [P_i(u_{i,j}) - P_{i,j}] \\ &= [P_i(u_{i,j}) - P_{i,j}]^2, \\ &\qquad i = 0,1,2,......n-1,\ j = 0,1,2,......, m+1, \end{aligned}$$

Due to interpolation conditions, we have

$$\begin{aligned} d_{i,j} &= [P_i(u_{i,j}) - P_{i,j}]^2, \\ &\qquad i = 0,1,2,......n-1,\ j = 1,2,......, m, \end{aligned} \quad (18)$$

where the parameterization over u's is in accordance with the chord length parameterization and is as follows:

$$\left.\begin{aligned} u_{i,0} &= t(i); \\ u_{i,j} &= u_{i,j-1} + |P_{i,j}P_{i,j+1}|, \\ &\qquad i = 0,1,2,......n,\ j = 1,2,.....m_i + 1 \end{aligned}\right\} \quad (19)$$

For each segment, if it is not as accurate as desired in the default spline case when $v_i = 3 = w_i$, we use the intermediate point method to improve the accuracy. We will look for an intermediate points to achieve the best optimal values of the corresponding shape parameters *v*'s and *w*'s. For the best fitting of the curve to the given data, we have to find out the parameters v_i's and w_i's so that the squared distances $d_{i,j}$'s are minimal. Obviously, we will look for those values of $u_{i,j}$'s in each interval $[t_i, t_{i+1}]$ for which the corresponding $d_{i,j}$'s are highest. Such intermediate points, searched in each interval, thus obtained will be desired characteristic points. The parameters v_i's and w_i's can be solved by the method given in Section.

4.2.2. Step 2

Now, redraw the curve using the Intermediate Point Interpolation method of Subsection 2.2. The spline curve fitted in this way will be a good candidate of best fit.

There is a possibility that the spline curve, obtained by the above process of Step 1, may not be satisfactory and error to the original data outline may not be as desired by the user. This error can be reduced by subdividing the segments at the intermediate points. Redraw the curve using the Intermediate Point Interpolation method of Subsection 2.2.

4.2.3. Step 3

After the Intermediate Point Interpolation method in Step 2, however, still there is a possibility that the spline curve obtained by the above process of Step 1 and 2, may not be satisfactory and error to the original data outline may not be as desired by the user. This error can be reduced by subdividing the segments at the intermediate points. Redraw the default spline curve method to the updated characteristic points.

4.2.4. Step 4

If the output is yet to be corrected, the Steps 2 and 3 are repeated as long as the desired output is achieved.

5. ALGORITHM DESIGN

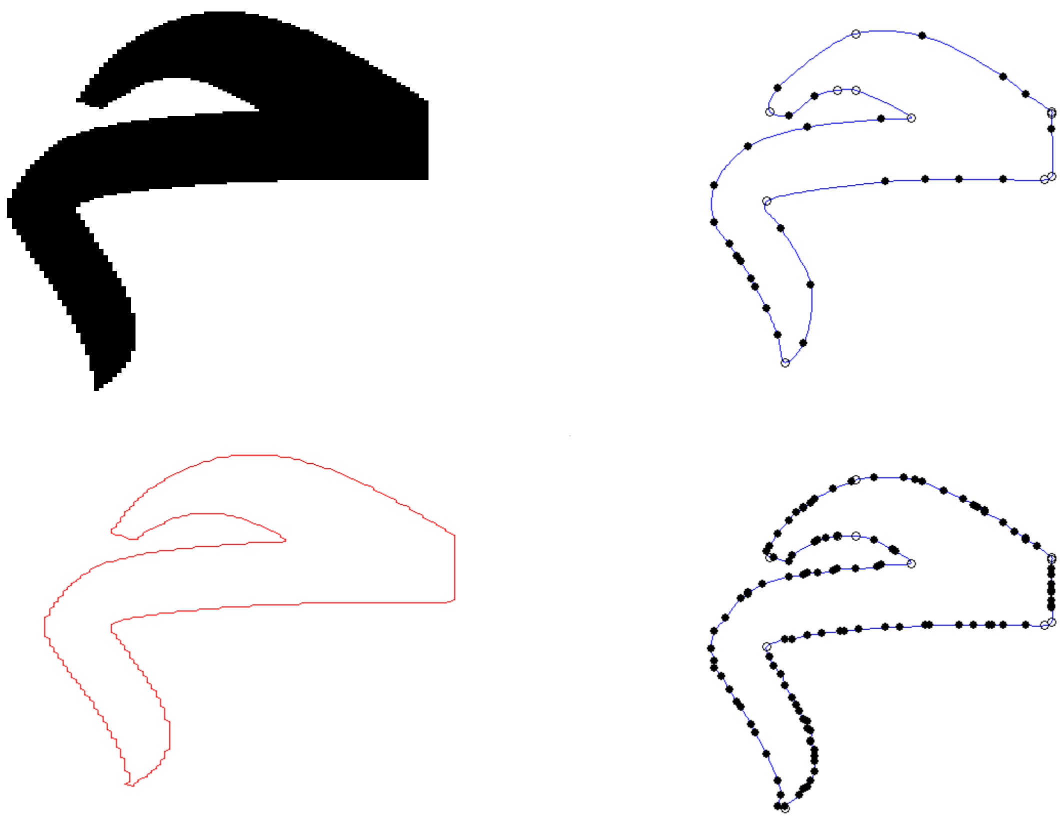
This section summarizes various steps discussed in this article. The desire of achieving an optimal fit, which inherits the shape of the data, leads to an algorithm discussed as follows.

Algorithm

Step 1: Input the data points.

Step 2: Subdivide the data, in Step 1, by computing the characteristic points using the method in Subsection 4.1.

Step 3: Compute the derivative values at the characteristic points.



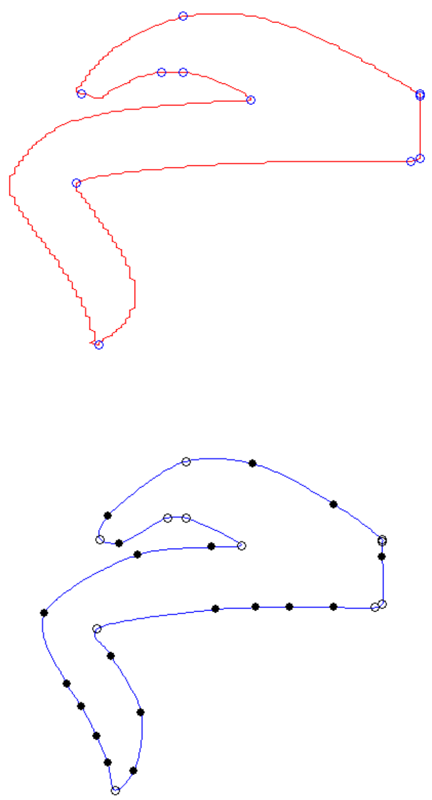


Figure 4. (a) The Arabic character "Meem", (b) Outline of Meem, (c) Detecting significant points on the outline, (d) The curve outline, with threshold value 3, showing the significant points (circles) and the characteristic points (bullets), (e) The curve outline with threshold value 2, (f) The curve outline with threshold value 1.

Step 4: Fit the spline curve method, of Section 3, to the characteristic points achieved in Step 2.

Step 5: If the curve, achieved in Step 4, is optimal then GO To Step 7, ELSE locate the appropriate intermediate points (points with highest deviation) in the undesired curve pieces. Compute the best optimal values of the shape parameters v_i's and w_i's. Fit the intermediate point interpolation curve method, of Section 2.

Step 6: IF the curve, achieved in Step 5, is optimal then GO To Step 7, ELSE enhance the list of characteristic points by incorporating the intermediate points (points with highest deviation) located in Step 5 and GO TO Step 3.

Step 7: STOP.

The above mentioned scheme and the algorithm have been implemented and tested for various data sets.

Reasonably quite elegant results have been observed, see the following Section for demonstration.

6. DEMONSTRATION

The algorithm has been implemented for the data obtained from the outline of a scanned image of the Arabic character “Meem”, see Fig.4(a). The edge detection method of [Che99] has been used for this purpose, see Fig.4(b). There are 467 data points in this outline. The significant points (circles) detected are shown in Fig.4(c). Figs.4(d), 4(e), and 4(d) are corresponding to threshold values 3, 2, and 1 respectively. The bullets, in Figs.4(d-f), are representing the intermediate points obtained for the best fit.

7. CONCLUDING REMARKS

A curve scheme, which initially generates an interpolation curve, has been proposed. The curve scheme is meant for design applications in various fields including Image Processing, Pattern Recognition, CAD/CAGD, and Computer Graphics. The scheme has a number of desirable properties. For example, it provides the setup in which the shape of the curve can be altered by the user. The curve can be modified locally with the help of shape control parameters as well as intermediate point interpolation technique. In this way, the scheme is quite simple, easy to implement and computationally economical.

Another feature of the scheme is that it provides the best optimal fit to a large amount of data. That is, the data can be visualized with an optimal spline curve fit with using a very little number of data points from the large input data. A case study of two examples has been made from the data collected for the outlines of fonts from two different languages.

The author is looking, as a future work, to extend the curve idea for the designing of surfaces. This could be quite useful for various applications in different fields of studies.

8. ACKNOWLEDGMENTS

The author is indebted to the anonymous reviewers for their constructive, valuable, and helpful comments. The author also acknowledges the support of King Fahd University of Petroleum and Minerals, against Project No. FT/2001-18, in the development of this work.

9. REFERENCES

[Avr91] Avrahami, G., and Pratt, V. Sub-Pixel Edge Detection in Character Digitization, Raster Imaging and Digital Typography II, Eds.. Morris, R. and Andre, J., Cambridge University Press, 54-64, 1991.

[Bro93] Brodlie, K.W., and Butt, S. Preserving Positivity using piecewise cubic interpolation, Comput. & Graphics, 17(1), 55-64, 1993.

[Che99] Chetverikov, D. and Szabo, Z. A simple and efficient algorithm for detection of high curvature points in planar curves. Proc. 23rd Workshop of the Australian Pattern Recognition Group, 175-184, 1999.

[Dav79] Davis, L. Shape Matching using Relaxation Techniques, IEEE Trans. PAMI, Vol. 1, 60-72, 1979.

[Far94] Farin, G. E. Curves and Surfaces for Computer Aided Geometric Design, Academic Press, New York, 1994.

[Gre86a] Gregory J. A. Shape preserving spline interpolation, Computer-Aided Design, 18(1), 53-57, 1986.

[Gre90b] Gregory J. A. and Sarfraz, M. A Rational Spline with Tension, Computer Aided Geometric Design, 7(1), 1-13, 1990.

[Hus89] Hussain, F. Conic Rescue of Bezier Founts, New Advances in Computer Graphics, Proceeding of CG International'89, Springer-Verlag, Tokyo, 97-120, (1989).

[Sar01a] Sarfraz, M., and Khan, M. A. Automatic Outline Capture of Arabic Fonts, Journal of Information Sciences, Elsevier Science Inc., 140(3-4), 269-281, 2001.

[Sar02b] Sarfraz, M. and Razzak, M. F. A. Automatic Font Outline Capturing Through WEB, The Proceedings of the Fourth KFUPM Workshop on Information and Computer Science: Internet Computing (WICS’2002), Saudi Arabia, ISBN: 9960-07-187-1, 237-251, 2002.

[Sar02c] Sarfraz, M. and Raza, A. Visualization of Data using Genetic Algorithm, *Soft Computing and Industry: Recent Applications*, Eds.: R. Roy, M. Koppen, S. Ovaska, T. Furuhashi, and F. Hoffmann, ISBN: 1-85233-539-4, Springer, 535-544, 2002.

[Sar02d] Sarfraz, M. Some Remarks on a Rational Cubic Spline for the Visualization of Monotonic Data, Computers & Graphics, Elsevier Science, Vol. 26(1), 193-197, 2002.