Capturing Outlines of Planar Generic Images by Simultaneous Curve Fitting and Sub-division

A. Ebrahimi¹, Gh. Barid Loghmani¹* and M. Sarfraz²

¹. Computer Geometry and Dynamical Systems Laboratory, Faculty of Mathematical Sciences, Yazd University, Yazd, Iran.
². Department of Information Science, Kuwait University, Kuwait.

Received 06 February 2018; Revised 23 January 2019; Accepted 02 March 2019
*Corresponding author: loghmani@yazd.ac.ir (Ghasem Barid Loghmani).

Abstract
In this work, a new technique is designed to capture the outline of 2D shapes using cubic Bezier curves. The proposed technique avoids the traditional method of optimizing the global squared fitting error, and emphasizes on the local control of data points. A maximum error is determined to preserve the absolute fitting error less than a criterion, and it administers the process of curve sub-division. Depending on the specified maximum error, the proposed technique itself sub-divides complex segments, and curve fitting is done simultaneously. A comparative study of the experimental results embosses various advantages of the proposed technique such as accurate representation, low approximation errors, and efficient computational complexity.

Keywords: Outline Capturing, Cubic Bezier Curve, Curve Fitting, Control Points, Curve Sub-division, Corner Points.

1. Introduction
Capturing the outline of 2D objects has been one of the prime challenges in computer-aided geometric design (CAGD), computer graphics, as well as vision and imaging [5, 11, 19]. Various mathematical and computational stages are associated with the whole process. Curve fitting has an important influence on this process. The outline representation of an object describes the object outlines as linear combinations of control points and blending functions. Outline representation preserves the complete shape of an object. The outline representation of 2D objects has various usefulness such as scaling up or down, translation, rotation, and clipping [21].

Researchers have introduced various techniques to represent outline objects. A vast majority of these techniques are according to least square curve fitting [10, 15, 27] and error optimization [3, 22]. These methods have complex computations; therefore, they cannot be used for real time applications. Itoh and Ohno [10] used cubic Bezier curves to approximate outline character fonts. They utilized the least square curve fitting to determine control points without fixing the end points of the curves. Sarfraz and Khan [21] employed least square fit for capturing the bitmap characters. Their method consists of several phases, such as ascertaining of edges, finding corner points and break points, and approximating the target shape. Sarfraz and Razzak [27] utilized generalized Hermite cubic spline to represent the outline of digital character images using characteristic points. Corner detection and sub-division method used in this scheme get very computationally expensive and sub-optimal. Thus, this algorithm has a low compression ratio. They also used control parameters to increase plasticity in curve fitting and imposed an additional computational time to find appropriate values for these parameters. Sarfraz and Khan [22] used cubic Bezier curves for approximating boundary and added reparameterization phases to ameliorate the efficiency of method. Large errors with complex and time-consuming computing are the detriment
of this method. The enhanced Bezier curve model presented by Sohel et al. [30] decreases the spacing between the fitting curve and its control polygon without additional computational complexity. Masood and Sarfraz [13] proposed an algorithm to represent outlines of planar images according to sub-division of outlines and optimal Bezier curve fitting. This approach determined control points by a search algorithm producing optimal curves. Sarfraz and Masood [23] presented an outline capturing scheme using cubic Bezier curves. This method determines the appropriate position of the control points by using the attributes of cubic Bezier curves. In [8], a local C1 positivity preserving process is extended using Bernstein Bezier rational cubic function. Sarfraz et al. [20] presented a C1 rational cubic piecewise function with two parameters. Hussain et al. [9] introduced a method to preserve the shape of data using a trigonometric interpolant. In [7], quadratic trigonometric B-spline with control parameters is created to consider the 2D image interpolation subject. Hasan et al. [6] compared cubic Bezier and cubic Ball curves using differential evolution to discover the best curve in curve reconstruction. A procedure has been proposed in [29] to construct B-spline like local support basis functions with a quadratic trigonometric function. Abbas et al. [1] constructed the rational ball cubic B-spline to interpolate natural images. They used genetic method to find the optimal value of tension parameter. Samreen et al. [16] introduced a quadratic trigonometric spline for object modeling. Their method has appropriate geometric properties such as partition of unity, convex hull, affine invariance, and variation diminishing. Some other techniques based on the search algorithm include evolutionary algorithm, simulated annealing approach, genetic algorithm, and wavelets. Non-uniform rational B-splines (NURBS) have been utilized in [28] to approximate the outlines of planar shapes using simulated annealing. Sarfraz [17, 18] used simulated annealing to optimize a cubic spline for outline capture of generic shapes. Sarfraz and Reza [25, 26] introduced an outline capturing scheme using genetic algorithm and splines. Tang et al. [31] introduced a method based on cubic B-spline wavelet transform to scale the Chinese character. The methodology presented in this work differs from the common methods in many aspects. In prevalent techniques, complex segments are subdivided into two pieces in a suitable position through analyzing the deviation error. In [21, 22, 23], the complex segment is partitioned into two pieces at the point of maximum deviation fitting error if the maximum deviation fitting error oversteps the given criterion. Masood and Sarfraz [12] analyzed the control points spread (CPspread) to sub-divide complex segments into two or more segments. In all the mentioned techniques, curve sub-division is an accessory step after the first fitting that is done to reduce the approximation error. This paper employs cubic Bezier curves to approximate the outline of 2D objects with a reasonable accuracy. The approach presented in this paper emphasizes the local control in lieu of optimizing the global least squared error. Our approach uses a maximum error to retain the absolute fitting error within a limit during curve fitting of a segment. The control points of the approximated curve are regulated through the properties of the cubic Bezier curves in lieu of the least square curve fitting or data point interpolation. The suggested method can also approximate outline segments and simultaneously sub-divide complex segments. In fact, curve fitting and curve sub-division are accomplished simultaneously. There are some features to distinguish our method from traditional approaches.

This paper is structured as what follows. Section 2 reviews the segmentation of an outline using corner detection. The process of determine control points for cubic Bezier curve approximation and the simultaneous sub-division of complex segments are explained in section 3. Experimental results and their comparison with other algorithms are demonstrated in Sections 4 and 5, respectively. Finally, we conclude the paper in Section 6.

2. Outline segmentation

The outline of 2D objects is mostly complex in shape. Therefore, boundary segmentation is used to divide the shape outline into small pieces and simplify the curve fitting procedure. Also, there are intrinsic break points in the shape that we desire to maintain as they are. We use the corner detection method to partition the outline into disparate pieces from the natural break points. Various algorithms have been suggested in literature for corner detection [3, 4, 24]. These algorithms do curvature analysis with numerical techniques. In this paper, the method proposed in [24] is adopted for the corner detection because this algorithm is accurate, affective and robust to noise. This algorithm is briefly explained in this section (readers are referred to [24] for details). The algorithm is designed in two phases.

In the first phase, candidate corner points are detected from the whole data set. If all the partition
data points are $Q_i, 1 \leq i \leq n$, the contour point $Q_k$ can be given as:

\[
\begin{align*}
\text{if} \quad (i + L) \leq n, \\
\text{then} \quad Q_k = Q_{i+L}, \\
\text{else} \quad Q_k = Q_{(i+L)-n},
\end{align*}
\]

where $L$ is a length parameter that describes the region of support and preserves the shape scaling and resolution. The default value for the length parameter is 14. There is a direct line connecting the points $Q_i(x, y)$ and $Q_k(x, y)$, and the perpendicular distance $h_j$ from point $Q_j(x, y)$ to this line can be given as follows:

\[
\begin{align*}
\text{if} \quad m_y = 0, \\
\text{then} \quad h_j = |Q_{j,y} - Q_{x,y}|, \\
\text{else} \quad h_j = \frac{|Q_{j,y} - mQ_{j,x} + mQ_{x,y} - Q_{x,y}|}{\sqrt{m^2 + 1}},
\end{align*}
\]

where $m = m_x = \frac{Q_{j,y} - Q_{x,y}}{Q_{j,x} - Q_{x,x}}$.

The point $Q_j$ is chosen as a volunteer corner point if its perpendicular distance ($h_j$) from the direct line $Q_jQ_k$ goes beyond $D$. $D$ is a distance parameter that considers the local sharpness and opening angle of the corner points. This parameter can control incorrect determination of corner points by cause of noise and disarray. The default value of $D$ is taken 2.6. Next candidate corner points are determined for a new direct through increasing both $i$ and $k$. In the second step, the superfluous corner points are detected. A superfluous corner point is one of that any other volunteer with a greater value of $h_j$ that is in the domain. The default value for $R$ is equal to length parameter. In this work, we utilized this method at its default values for boundary segmentation.

3. Curve approximation

The outline of objects is partitioned into curve segments based on corner points. It means that all the boundary points between two consecutive corner points constitute one curve segment. In this work, we applied a cubic Bezier curve [2, 32] for curve fitting in each segment separately.

The Bezier polynomial is a parametric curve $C(\alpha)$ widely used in computer graphics and shape modeling. The degree of the Bezier curve appertains to the number of control points exerted to describe the curve. A Bezier polynomial with a degree of $n - 1$ has $n$ control points. The Bezier polynomial pass through the two end control points $P_0$ and $P_n$ are attracted by the middle control point $P_1, \ldots, P_{n-1}$.

A cubic Bezier curve is given as:

\[
C(\alpha) = P_0G_{0,3}(\alpha) + P_1G_{1,3}(\alpha) + P_2G_{2,3}(\alpha) + P_3G_{3,3}(\alpha), \quad 0 \leq \alpha \leq 1.
\]

The blending functions $G_{0,3}, G_{1,3}, G_{2,3}$ and $G_{3,3}$ are cubic Bernstein polynomials, which are defined as follow:

\[
\begin{align*}
G_{0,3}(\alpha) &= (1-\alpha)^3, \\
G_{1,3}(\alpha) &= 3\alpha(1-\alpha)^2, \\
G_{2,3}(\alpha) &= 3\alpha^2(1-\alpha), \\
G_{3,3}(\alpha) &= \alpha^3.
\end{align*}
\]

Figure 1 shows the family of the four cubic Bernstein polynomials.

It is straightforward to show that cubic Bezier curves have a variety of useful properties; most of the results come directly from the properties of the Bernstein polynomials [14].

- Partition of unity: $\sum_{i=0}^{3} G_{i,3}(\alpha) = 1$, for all $0 \leq \alpha \leq 1$.
- Affine invariance.
- Convex hull property.
- End-point interpolation.
- $G_{1,3}(0.5) = G_{2,3}(0.5)$.

- Value of Bernstein polynomials, for all $0 \leq \alpha \leq 1$ does not appertain to the position of any control point(s) [12].
- If the position of any control point is specified, its efficacy all over the curve can be deprived [12].
In this section, we use the properties of Bezier curves to estimate intermediate control points and fit the best optimal curve so as to simultaneously sub-divide the complex segments. Suppose \( \{q_1, q_2, \ldots, q_m\} \) are given an ordered set of the contour point of a segment. A cubic Bezier curve for fitting these points is determined by the four control points \( P_0, P_1, P_2, P_3 \). The cubic Bezier curve begin from the control point \( P_0 \) and terminate at the control point \( P_3 \). Thus, the effect of \( P_0 \) and \( P_3 \) is removed from a defined cubic Bezier polynomial as:

\[
P G_{1,3} (\alpha) + P_2 G_{2,3} (\alpha) = C_3' (\alpha),
\]

Where:

\[
C_3' (\alpha) = C_3 (\alpha) - P_2 G_{0,3} (\alpha) - P_3 G_{1,3} (\alpha).
\]

Based on the properties of the cubic Bezier curves, we get:

\[
G = G_{1,3} (0.5) = G_{2,3} (0.5).
\]

From Equations (2) and (4), we have:

\[
P_1 + P_2 = C_3' ,
\]

Where:

\[
C_3'' = \frac{C_3' (0.5)}{G}.
\]

By solving Equations (2) and (5), we have:

\[
\begin{cases}
P_1 = \frac{C_3' (\alpha) - G_{2,3} (\alpha) C_3''}{G_{1,3} (\alpha) - G_{2,3} (\alpha)}, \\
P_2 = C_3' - P_1.
\end{cases}
\]

The positions of the intermediate control points \( P_1 \) and \( P_2 \) remain constant for the cubic curve throughout the interval \( \alpha \) \( 0 \leq \alpha \leq 1, \alpha \neq 0, 0.5, 1 \) [12]. For non-cubic curves, the positions of the control points \( P_1 \) and \( P_2 \) may not remain constants and these points spread around the curve.

Let \( P_1^i \) and \( P_2^i \) be the computed intermediate control points at \( \alpha_i \), \( 0 \leq \alpha_i \leq 1, \alpha_i \neq 0, 0.5, 1 \)

\[
\begin{align*}
P_1^i &= C_3' (\alpha_i) - G_{2,3} (\alpha_i) C_3'' \\
&= \frac{G_{1,3} (\alpha_i) - G_{2,3} (\alpha_i)}{G_{1,3} (\alpha_i) - G_{2,3} (\alpha_i)}.
\end{align*}
\]

\[
P_2^i = C_3' - P_1^i.
\]

Let us assume \( m - 2 \) cubic Bezier curve for the presentation of \( m \) data points \( \{q_1, q_2, \ldots, q_m\} \) such that:

\[
q_i = C_3^i (t_i), \quad i = 2, \ldots, m - 1.
\]

where \( C_3^i (\alpha_i) \) is the value of the \( i \) th cubic Bezier curve at the point \( \alpha_i \). It is given by:

\[
C_3^i (\alpha_i) = P_0 G_{0,3} (\alpha_i) + P_1^i G_{1,3} (\alpha_i) + P_2^i G_{2,3} (\alpha_i) + P_3 G_{3,3} (\alpha_i), \quad 0 \leq \alpha_i \leq 1.
\]

Inserting (8) into (9), we get:

\[
C_3^i (\alpha_i) = P_0 G_{0,3} (\alpha_i) + P_1^i G_{1,3} (\alpha_i) + (C_3'' - P_1^i) G_{2,3} (\alpha_i) + P_3 G_{3,3} (\alpha_i).
\]

Let \( P_{\bar{1}} \) indicate the average value of the intermediate control points \( \{P_1^i\} \). From Equation (8), we have:

\[
\bar{P}_2 = C_3'' - \bar{P}_1.
\]

Let the corresponding cubic Bezier curve with the control points \( P_0, P_{\bar{1}}, \bar{P}_2 \) and \( P_3 \) be \( C_3 (\alpha_i) \).

The discrete form of \( C_3 (\alpha_i) \) can be written as:

\[
C_3 (\alpha_i) = P_0 G_{0,3} (\alpha_i) + \bar{P}_1^i G_{1,3} (\alpha_i) + (C_3'' - \bar{P}_1^i) G_{2,3} (\alpha_i) + P_3 G_{3,3} (\alpha_i).
\]

From Equations (10) and (11), we have:

\[
\| C_3^i (\alpha_i) - C_3 (\alpha_i) \| = \| (P_1^i - \bar{P}_1^i) G_{1,3} (\alpha_i) - (P_1^i - \bar{P}_1^i) G_{2,3} (\alpha_i) \|.
\]

\[
= \| (P_1^i - \bar{P}_1^i) (G_{1,3} (\alpha_i) - G_{2,3} (\alpha_i)) \|.
\]

where \( \|. \| \) denotes the Euclidean norm. Equation (12) denotes the difference of \( C_3 (\alpha_i) \) and \( C_3^i (\alpha_i) \) at the instant \( \alpha_i \).

The maximum difference of \( C_3 (\alpha_i) \) and \( C_3^i (\alpha_i) \) at interval \((0,1)\) is:
\[ \Box C_i^3(\alpha_i) - \Box C_i^3(\alpha_i) \leq e_{max} \]
\[ = \Box P_i^1 - \Box P_i^1 \leq 0.2886 \times G_{1,3}^1(\alpha_i) - G_{2,3}^2(\alpha_i) \]
\[ \leq 0.2886. \]

where \[ \Box C_i^3(\alpha_i) - \Box C_i^3(\alpha_i) = e_{max} \]
refers to the maximum errors in the approximation data points. From Equation (13), given any such number as
\( e_{max} > 0 \), there is another number \[ \delta = \frac{e_{max}}{0.2886} \]
such that
\[ \Box P_i^1 - \Box P_i^1 \leq \delta, \quad (14) \]
guarantees that:
\[ \Box C_i^3(\alpha_i) - \Box C_i^3(\alpha_i) \leq \Box C_i^3(\alpha_i) - \Box C_i^3(\alpha_i) \leq e_{max} \]
\[ \leq e_{max}. \]

Therefore, inequality (14) suggests that the data points \( q_i, i = 2, \ldots, m - 1 \) can be fitted by \( \Box C_i^3(\alpha) \)
with an error represented in inequality (15).

\textbf{Algorithm 1.} The outline capturing system.

\begin{itemize}
  \item Get digitized image and choose a maximum error (\( E_{max} \)).
  \item Extract outline;
  \item Detect corner points and divide the outline into segments;
  \item for each segment do
  \item \hspace{1cm} while all the points are approximated do
  \item \hspace{2cm} Fit the cubic Bezier curve to data points;
  \item \hspace{1cm} Compute \( P_i^1 \) and \( P_i^2 \);
  \item \hspace{1cm} Compute \( \delta = \frac{E_{max}}{0.2886} \);
  \item \hspace{1cm} if \( \Box P_i^1 - \Box P_i^1 \leq \delta \) then
  \item \hspace{2cm} break
  \item \hspace{1cm} else
  \item \hspace{2cm} Find the number of points that \( \Box P_i^1 - \Box P_i^1 \leq \delta \) is established
  \item \hspace{2cm} for them and approximate remaining points again;
  \item \hspace{1cm} end if
  \item \hspace{1cm} end while
  \item \hspace{1cm} end for
\end{itemize}

It is seen that inequalities (14) and (15) can be used to fit a segment of the planar shape. During curve fitting, it may be seen that inequality (14) does not maintain for all values of \( i \) related with a segment. Let the inequality (14) hold for \( m_0 \) points out of \( m \) in the segment. Thus, the remaining \( (m - m_0) \) points will again be fitted on the interval \([0,1]\). Therefore, the proposed method fits the cubic Bezier curve to a segment, and simultaneously subdivides complex segments. All steps of the outline capturing system is shown in Algorithm 1.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{figure2.png}
\caption{Fitting a non-parametric curve (black) using a cubic Bezier curve (red) with the proposed algorithm.}
\end{figure}

In order to illustrate the curve fitting with subdivision, we consider a non-parametric curve as shown in figure 2(a). The maximum error, \( e_{max} \), for curve fitting is 4 in figure 2(b). The fitting process
separates the curve into two pieces: the start and terminal points of each piece are fitted with zero error, whilst all the other data points are fitted with an error lower than $\varepsilon_{\text{max}} = 4$. In figure 2(c), the maximum error, $\varepsilon_{\text{max}}$, for curve fitting is 2. The fitting process separates the curve into three segments, and all the points are approximated with an error lower than $\varepsilon_{\text{max}} = 2$. In figure 2, the black and red lines indicate the original and approximated curves, respectively. The subdivided points are marked by *, and the intermediate control points are marked by •.

4. Experimental results

The implementation illustration of the presented approach is provided in this section. The aim of the proposed algorithm is to provide an accurate representation of an outline using an optimal number of segments without getting into complex computations. The results are analyzed, and the efficiency of the algorithm is evaluated for different $\varepsilon_{\text{max}}$ on the basis of the parameters below.

- Number of segments: The outline is partitioned into a number of segments using a corner detection algorithm, and, during curve fitting, complex segments are divided into several segments. An increase in the number of segments, leads to an increase in the number of extracted data points. Therefore, a minimum number of segments are favorable.

- Compression ratio (CR): This is a significant criterion that represents an evaluation about the quantity of compression accomplished by the method. A large compression ratio is desired. This criterion indicates the ratio of the number of data points in actual outline ($n$) to the obtained data points in curve fitting ($n_{\text{DP}}$). It is calculated by the following equation:

$$CR = \frac{n}{n_{\text{DP}}}. \quad (16)$$

- Average error: It refers to all errors generated in the approximated outline, and is defined by:

$$\text{Average error} = \frac{1}{n} \sum_{i=1}^{n} e_i \quad (17)$$

Where $e_i$ is the Euclidean distance between the $i$th point of the outline and the corresponding point of the parametric curve.

- Maximum deviation: This criterion computes the maximum deviation of the approximated outline from the primary shape. It can be measured by:

$$\text{Maximum deviation} = \max_{i=1}^{n} \{e_i\} \quad (18)$$

- Computational time: It is the quantity of time needed to implement the method and appertain to the performance approach and the processor used for implementing the method.

The algorithm has been tested for two characters in figures 3 and 4. Figure 3(a) shows the original shape of an Arabic word "Sabr"; and the extracted outline is given in Figure 3(b). Figure 3(c) displays the corner points that are denoted by a square. Curve fitting with the proposed approach is implemented over segmented outlines, and the results obtained are shown in figures 3(d-f). The maximum error, $\varepsilon_{\text{max}}$, has been preserved as 2.5, 2 and 1.75 for figures 3(d), 3(e), and 3(f), respectively. The corner and sub-division points are marked by a square and a circle, respectively. The detailed quantitative results in Figure 3 can be seen in table 1. One can see that the approximated boundary in figure 3(d) with a maximum error of 2.5 is quite reasonable and quite close to the original boundary. However, maximum errors of 2 and 1.75 provide better results in figures 3(e) and 3(f); however, they impose a higher cost in terms of computation, and decrease the compression ratio. Another example with the proposed method, similar to that in figure 3, has been achieved for Kanji character in figure 4. Detailed quantitative results are shown in table 2. One can see again that the approximated boundary in figure 4(d) with a maximum error of 2.5 is quite reasonable and quite close to the original boundary. Although a less maximum error value of 2.5 provides more accurate results in figures 4(e) and 4(f), it raises the number of segments and causes a higher cost in terms of computation.

5. Comparative study

In order to illustrate the effectiveness of the proposed algorithm, its results are hereby compared with those of algorithm [12, 22, 27]. A visual comparison and detailed quantitative results are employed to highlight the advantages of the proposed method such as accurate representation, low average error, low computational time, and high compression ratio.

A comparison is first made of the proposed method and the two methods [12, 27] for the outline of the Arabic word "Sabr", as shown in figures 3(a) and 3(b). The visual comparison of these algorithms is shown in figure 5. The corner and sub-division
points are denoted by a square and a circle respectively. Table 3 provides the comparative results for this example. The result of the proposed method with a maximum error of 2.5 is given in figure 5(a). Here, the approximation is performed with 28 segments. The maximum deviation and the average error procured are 2.29 and 0.64, respectively. The outline captured by Masood and Sarfraz [12] is shown in figure 5(b). In this algorithm [12], the shape is computed with 27 segments and a maximum deviation of 2.21, which are less than those generated by the proposed approach. However, the values of the average error and the computational time are higher, i.e. 1.38 and 0.83 respectively. Capturing outline schemes for 2D objects with algorithm [27] at a criterion of 3 and 2 is depicted in figures 5(c) and 5(d), respectively. The number of segments at the criterion of 3 and 2 is 66 and 98, respectively, which is very high when compared with that of the proposed method. The cause for the high number of segments is the use of sub-optimal corner detection and sub-division method. The inappropriate corner finding and sub-division method in the case of this algorithm are pointed out by arrows in figures 5(c) and 5(d).

The computational time at the criterion of 3 is 3.79, which is about 4.80 times more than that generated by the presented method. Likewise, the computational time at the criterion of 2 is 2.58, which is about 3.27 times more than that generated by the presented method. The least squares fitting and the employment of control parameters to expand plasticity in curve fittings are the two causes of the high computational time in this method. The average error at the criterion of 3 and 2 is 1.07 and 0.91, respectively, which is higher than that of the proposed method. The maximum deviation with algorithm [27] at both criterions (i.e. 2 and 3) is lower than that of the presented method but this low error is at the cost of very high segments and computational time.

### Table 1. Performance of the proposed algorithm for the shape of the Arabic word "Sahr".

<table>
<thead>
<tr>
<th>$\varepsilon_{max}$</th>
<th>Number of segments</th>
<th>Compression ratio</th>
<th>Maximum deviation</th>
<th>Average error</th>
<th>Computation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>28</td>
<td>46.43</td>
<td>2.29</td>
<td>0.64</td>
<td>0.79</td>
</tr>
<tr>
<td>2</td>
<td>33</td>
<td>39.39</td>
<td>1.92</td>
<td>0.53</td>
<td>0.93</td>
</tr>
<tr>
<td>1.75</td>
<td>35</td>
<td>37.14</td>
<td>1.74</td>
<td>0.50</td>
<td>1.27</td>
</tr>
</tbody>
</table>

### Table 2. Performance of the proposed algorithm for the shape Kanji character.

<table>
<thead>
<tr>
<th>$\varepsilon_{max}$</th>
<th>Number of segments</th>
<th>Compression ratio</th>
<th>Maximum deviation</th>
<th>Average error</th>
<th>Computation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>33</td>
<td>64.80</td>
<td>2.18</td>
<td>0.66</td>
<td>0.53</td>
</tr>
<tr>
<td>2</td>
<td>36</td>
<td>59.40</td>
<td>1.91</td>
<td>0.56</td>
<td>0.80</td>
</tr>
<tr>
<td>1.75</td>
<td>42</td>
<td>50.92</td>
<td>1.43</td>
<td>0.49</td>
<td>0.95</td>
</tr>
</tbody>
</table>
Figure 3. Captured shape of the Arabic word "Sabr". The black and red lines show the original and computed curves, respectively.
Figure 4. Captured shape of Kanji character. The black and red lines show the original and computed curves, respectively.
The capturing results for the original outline of Kanji character are shown in figure 6 and tabulated in table 4. The outline captured using the proposed method with a maximum error of 2.5 is given in figure 6(a). Capturing of the shape has been carried out with 33 segments, and the shape has been computed within 0.53 s. The maximum deviation is 2.18, and the average error is 0.66. The outline computed using method [12] is shown in figure 6(b). In this method [12], the number of segments is the same as that of the proposed method. The maximum deviation, i.e. 1.55, is less than that in the proposed method, however both the average error and the computational time are higher, with values of 0.73 and 0.59, respectively. Another capturing result is given in figure 6(c) using algorithm [22]. In this algorithm, the approximation is performed with 31 segments, which is fewer than that generated by the proposed method. The maximum deviation and the average error, however, are 3.78 and 1.92, respectively, which are about 1.73 and 2.91 times higher than those in the proposed method. The noise filtering process before computational is the basic reason of the high error. The noise filtering also causes inaccuracy of the computed shape. It is shown with an arrow in figure 6(c). The causes of the high computational time 3.61 are the least squares curve fitting and the procedure of noise filtration. Another capturing result is given in figure 6(d) using algorithm [27]. The segments used in this case are double as many as those taken by the proposed method. The computational time with this algorithm is 3.83, which is about 7.3 times higher than that in the presented method. The causes of high computational time are already mentioned.

Based on the comparative study, we found the following advantages for the proposed method:

- The proposed algorithm controls a maximum error for each piece in lieu of optimization the global squared sum of errors.
- Curve fitting and curve sub-division are accomplished simultaneously.
- The proposed method provides an efficient curve approximation in which the average error and the computational time reduce considerably (as compared to other methods in the current literature [12, 27]).
- The proposed method does not assure an optimal solution but it preserves an appropriate equilibrium between the compression ratio and the fitting error.

Table 3. Results of quantitative performance with different algorithms for the outline of Arabic word "Sabr".

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Number of segments</th>
<th>Compression ratio</th>
<th>Maximum deviation</th>
<th>Average error</th>
<th>Computation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed ($\varepsilon_{max} = 2.5$)</td>
<td>28</td>
<td>46.43</td>
<td>2.29</td>
<td>0.64</td>
<td>0.79</td>
</tr>
<tr>
<td>Masood and Sarfraz [12]</td>
<td>27</td>
<td>48.15</td>
<td>2.21</td>
<td>1.38</td>
<td>0.83</td>
</tr>
<tr>
<td>Sarfraz and Razzak [27] at $\tau = 3$</td>
<td>66</td>
<td>19.70</td>
<td>1.71</td>
<td>1.07</td>
<td>3.79</td>
</tr>
<tr>
<td>Sarfraz and Razzak [27] at $\tau = 2$</td>
<td>98</td>
<td>13.27</td>
<td>1.39</td>
<td>0.91</td>
<td>2.58</td>
</tr>
</tbody>
</table>
Table 4. Results of quantitative performance with different algorithms for the shape of Kanji character.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Number of segments</th>
<th>Compression ratio</th>
<th>Maximum deviation</th>
<th>Average error</th>
<th>Computation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed ( $\varepsilon_{max} = 2.5$ )</td>
<td>33</td>
<td>64.81</td>
<td>2.18</td>
<td>0.66</td>
<td>0.53</td>
</tr>
<tr>
<td>Masood and Sarfraz [12]</td>
<td>33</td>
<td>64.81</td>
<td>1.55</td>
<td>0.73</td>
<td>0.59</td>
</tr>
<tr>
<td>Sarfraz and Khan [22]</td>
<td>31</td>
<td>69</td>
<td>3.78</td>
<td>1.92</td>
<td>3.61</td>
</tr>
<tr>
<td>Sarfraz and Razzak [27]</td>
<td>66</td>
<td>33.42</td>
<td>1.40</td>
<td>1.13</td>
<td>3.83</td>
</tr>
</tbody>
</table>

Figure 5. Captured shape of the Arabic word "Sabr" with different methods.

(a) Proposed method with $\varepsilon_{max} = 2.5$.
(b) Masood and Sarfraz [12].
(c) Sarfraz and Razzak [27] at $\tau = 3$.
(d) Sarfraz and Razzak [27] at $\tau = 2$. 
6. Conclusion

An outline capturing system has been presented based on the cubic Bezier curve. The control points of the approximated shape are regulated through the attributes of the cubic Bezier polynomials. The introduced approximation technique has advantages such as low average error, low computational time, and high compression ratio. The proposed algorithm emphasizes the positional control of segments in lieu of optimization the global squared error. A maximum error has been determined to preserve the absolute error less than a criterion. Depending on the specified maximum error, the proposed technique itself sub-divides complex segments, and approximation is done simultaneously. The performance of algorithms is practically based on 2D images of different shapes. The proposed algorithm has various features such as proper shape preservation, desirable equivalency between compression ratio and fitting error, low average error, and low computational time. These specifications have been authenticated by comparing the proposed method with a few other methods. The method can be fruitful in geometric modelling, font designing, and computer aided-geometric design (CAGD).
References


استخراج پیرامون تصاویر عمومی با استفاده از برآزش منحنی و قطعه بندی همزمان

علیرضا ابراهیمی، قاسم برید لقمانی و محمد سرافراز

1. آزمایشگاه هندرسه کامپیوتری و سیستم‌های دینامیکی، دانشکده علوم رياضی، دانشگاه یزد، یزد، ایران.
2. دانشکده علوم اطلاعات، دانشگاه کویت، کویت.

ارسال: 60/60/2006، بازخوانی: 60/62/2003، پذیرش: 1/12/2013

چکیده:
در این مقاله، یک روش برای استخراج پیرامون اشکال دو بعدی با استفاده از منحنی‌های بیزه درجه سطح استفاده است. روش بیشتری از روش‌های رایج برآزش منحنی و قطعه بندی با کاهش خطای برازش است. روش واقعی برآزش منحنی و قطعه بندی داده‌ها انجام داده و خطای روزهای بیشتری از روش‌های رایج را نشان می‌دهد. روش پایدار تر از روش‌های رایج است. محاسبات مناسب را نشان می‌دهد.

کلمات کلیدی: استخراج پیرامون، منحنی بیزه درجه سوم، برآزش منحنی، نقطه کنترل، قطعه بندی منحنی، نقطه گوشه.