Handwritten Curve Approximation by a Bézier Curve with Featured Points

Rattikarn Jaroensawad\textsuperscript{1}, Natasha Dejdumrong\textsuperscript{2}, Somchai Prakancharoen\textsuperscript{1}

Department of Computer and Information Science
King Mongkut’s University of Technology North Bangkok
Bangkok, Thailand\textsuperscript{1}

Department of Computer Engineering
King Mongkut’s University of Technology Thonburi
Bangkok, Thailand\textsuperscript{2}

{rattikarn_j@hotmail.com, natasha@cpe.kmutt.ac.th, spk@kmutnb.ac.th}

Abstract—Approximating a hand-written curve by a Bézier curve can be performed by two main processes: (1) the selection of the significant points on the curve and (2) the curve approximation. This paper focuses on finding a set of featured points, which is an important component for curve fitting into a Bézier curve. This new approach is an extension of the authors’ previous work by reducing the weakness of point sampling. In performance evaluation, a new point-sampling algorithm is used to compare with the previous methods which are the results from the approximation of handwritten curves by Chebyshev polynomials.

Keywords—hand-written curve approximation; featured point; Interpolation

I. INTRODUCTION

Curve fitting representation can be either an approximate function or new data points within a discrete set of known data points such as Linear, Quadratic and Polynomial Interpolation. Popular polynomial interpolations are Newton, Lagrange and Chebyshev polynomials, which can be commonly found in the books about Numerical Analysis and Computer Aided Geometric Design (CAGD). Although the approximate curve with polynomial fitting is not new or exciting, some attempts have been done on the improvement the research about handwritten curve approximation [2,3,5,6].

Today, there are several kinds of computer software that support for hand-drawing. Almost images from these programs are displayed in the forms of raster graphics. When these images are going to be further used, it will reach to some limitations or drawbacks such as contour smoothness or noise. This problem can be solved by reconstructing a new image, expressed in a vector graphics format. This can guarantee smoothness. In addition, it provides the flexibility in scaling up or even be applied for further work.

Recent researches [2, 3] have introduced two different featured point selection techniques for approximating handwritten curve. Both methods possess their disadvantages. For point selection by frequency, it likely avoids picking up some key points. Similarly, the point selection technique by the inflection point analysis, it is possible to choose some rubbish points. This leads to the higher error-correctness value. Therefore, the improved method in this study should avoid these drawbacks and focus on the importance of sampling points.

This research focuses on finding the featured points adopted to approximate the curve. The new results will be compared to those of previous work. Finally, the discussions and conclusions will be described in the last section.

II. RELATED THEORY

A. Polynomial

Polynomial is a mathematical expression, which is written in the form of sum of coefficient multiplied with the monomial as follows

\[ a_0 t^n + \ldots + a_j t^j + a_t + a_0, \]

where \( n \) is the highest degree of a polynomial, and \( a_0, a_1, \ldots, a_j, a_0 \) are constant coefficients.

There are several classes of widely used polynomials used interpolation.

1) Chebyshev Polynomial: A set of orthogonal polynomials whose values are within the range of \([-1,1]\). There are two important kinds of Chebyshev polynomials: the first kind of Chebyshev polynomials \( T_n(t) \) and the second kind of Chebyshev polynomials \( U_n(t) \). Both are the most popular. In this paper, only the first kind [7] are taken into account. The relation can be defined by

\[ T_n(t) = \cos n\theta, \]

where

\[ t = \cos \theta. \]
From the above equation, the fundamental recurrence relation can be obtained by
\[ T_n(t) = 2T_{n-1}(t) - T_{n-2}(t), \quad n = 2, 3, \ldots \quad (4) \]
In this work, a given curve will be transformed into a Bézier curve ranging from 0 to 1. Thus, the principle of shifted Chebyshev polynomials will be used for domain transformation. The formula can be defined by
\[ T_n(t) = T_n(2t - 1) \quad (5) \]
where the range of variable \( t \) is in \([0, 1]\).
Hence, the recursive function of Chebyshev polynomial can be rewritten as follows:
\[ T_n(t) = 2(2t - 1)T_{n-1}(t) - T_{n-2}(t) \quad (6) \]
The coefficients of Chebyshev polynomial can be calculated by the following equation
\[ c = y^T T_n^T, \quad (7) \]
where
- \( c \) is a coefficient matrix of Chebyshev polynomial,
- \( y \) is a matrix of sampling point, and
- \( T_n^T \) is a matrix of Chebyshev polynomial.

2) Bernstein Polynomial: Polynomials are very useful in many applications and were commonly used for constructing Bézier curves, parameterized by the interval \( t \in [0, 1] \). In general, Bernstein polynomials of degree \( n \) are defined by
\[ B_n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad (8) \]
where
\[ \binom{n}{i} = \frac{n!}{i!(n-i)!}, \quad i = 0, 1, \ldots, n. \quad (9) \]
The power basis of Bernstein functions can be rearranged into a new form by
\[ \sum (-1)^{i-k} \binom{n}{i} \binom{i}{k} t^i \]
where the binomial theorem to expand \((1-t)^{n-i}\), see more in [1].

B. Bézier Curve

Bézier curve is a parametric curve, frequently used in computer graphics and related fields. Bézier curves can be constructed by Bernstein polynomials. Besides, it can be defined by a set of control points \((b_0, b_1, \ldots, b_n)\), illustrated by
\[ B(t) = b_i B_n^i(t) = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} b_i \quad (11) \]
where \( B_n^i(t) \) is the Bernstein polynomial

For the sake of simplicity, it can be rewritten in the monomial form [4] as follows:
\[ B(t) = \begin{bmatrix} m_{0,0} & m_{0,1} & \cdots & m_{0,n} \\ m_{1,0} & m_{1,1} & \cdots & m_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,0} & m_{n,1} & \cdots & m_{n,n} \end{bmatrix} \begin{bmatrix} t^0 \\ t^1 \\ \vdots \\ t^n \end{bmatrix}, \quad (12) \]
where
\[ m_{i,j} = (-1)^{i-j} \binom{n}{j} \binom{j}{i}. \quad (13) \]

C. Transformation

1) Conversion from Chebyshev into Bézier curve: The formula can be presented in terms of a linear combination of the Bernstein polynomial basis [8] as follows
\[ T_n(t) = (2n-1)! \sum_{k=0}^{\min(j,k)} (-1)^{n-k} \binom{2k}{n-k} \binom{k}{j} B_n^k(t) \quad (14) \]
Moreover, the relationships between Chebyshev and Bernstein bases can be explicitly shown as follows
\[ P_n(t) = \sum_{i=0}^{n} b_i B_n^i(t) = \sum_{i=0}^{n} c_i T_n^i(t), \quad (15) \]
where
- \( b \) is the coefficients of Bernstein polynomial or Bézier control points, and
- \( c \) is the coefficients of Chebyshev polynomial.

Then it can readily be converted from Chebyshev coefficients into Bernstein coefficients by
\[ b_j = \sum_{k=0}^{n} M_{jk} c_k, \quad (16) \]
where
- \( M \) is the transformation matrix from the Chebyshev into the Bernstein bases, which is
\[ M_{jk} = \frac{1}{n} \sum_{i=\max(j,k-n)}^{\min(j,k)} (-1)^{n-k} \binom{2k}{i} \binom{2n-k}{j-i}. \quad (17) \]

D. Related Research

In work [3] presented two algorithms approximately handwritten curve by using technique of Chebyshev
Interpolation. Starting from computing the coefficients of Chebyshev polynomials in [7], we can adjust an intermediate Chebyshev curve before the transformation from the Chebyshev coefficients into the Bernstein coefficients by the following algorithm:

\[
\begin{align*}
\text{For } i = 1: i \leq \frac{n+1}{2}: i += 1, \\
& \{ a = \text{find error} \ t[i]; \\
& b = \text{find error} \ t[i] + 0.01; \\
& c = \text{find error} \ t[i] - 0.01; \\
& g = \min[a, b, c]; \\
& \text{If } g = a, \text{return} \ g; \\
& \text{Elseif } g = b \\
& \quad \text{for } j = g : j < t[i+1]: j = j + 0.01 \\
& \quad \quad h = \min[\text{find error} \ t[j]]; \\
& \quad \text{return} \ h; \\
& \text{Else} \\
& \quad \text{for } j = g : j > t[i]: j = j - 0.01 \\
& \quad \quad h = \min[\text{find error} \ t[j]]; \\
& \quad \text{return} \ h; \\
\} \\
\text{Change} \\
\text{For } i = n-1: i \geq \frac{n+1}{2}: i -= 1
\end{align*}
\]

Moreover, both algorithms are proposed for finding all possible featured points. The first algorithm uses technique of choosing sampling points by frequency and the second algorithm uses technique of selecting a set of sampling points by analyzing inflection points.

The results from both algorithms are acceptable and reliable. However, they contain some disadvantages, the first algorithm likely avoids picking up some necessary points. The illustration can be seen in figure 2 (left) and the second algorithm may pick up a number of rubbish points. It can be depicted in figure 2 (right).

![Fig. 2. Example of pick up featured point indecent.](image)

E. Geometric Transformation

In two-dimensional geometric transformations [5], there are three basic transformations: translation, rotation, and scaling.

1) **Translation**: Let \( t_x \) be the displacement in \( x \)-axis and \( t_y \) be the displacement in \( y \)-axis, translation can be computed by the following transformation matrix:

\[
\begin{bmatrix}
1 & 0 & t_x \\
0 & 1 & t_y \\
0 & 0 & 1
\end{bmatrix}
\]

The formula can be expressed by

\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & t_x \\
0 & 1 & t_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
\]

(19)

2) **Rotation**: Given the angle \( \theta \) for the rotation to turn the shape around a pivot point. It is normally fixed to the origin. It can be computed by the following transformation matrix:

\[
\begin{bmatrix}
\cos \theta & -\sin \theta & t_x \\
\sin \theta & \cos \theta & t_y \\
0 & 0 & 1
\end{bmatrix}
\]

(20)

The formula can be shown by

\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} =
\begin{bmatrix}
\cos \theta & -\sin \theta & t_x \\
\sin \theta & \cos \theta & t_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
\]

(21)

3) **Scaling**: Given \( s_x, s_y \) the scale factors in the \( x \)-axis and \( y \)-axis. It can be computed by the following transformation matrix:

\[
\begin{bmatrix}
s_x & 0 & 0 \\
0 & s_y & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

(22)

The conversion equation is formulated by

\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} =
\begin{bmatrix}
s_x & 0 & 0 \\
0 & s_y & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
\]

(23)

Remarks: For rotation and scaling, if pivot point or fixed point about an arbitrary axis, we have to first translate a point on this axis into the origin before performing the transformation.
III. PROPOSED METHODOLOGY

In general, the technique for curve approximation should be simple and fast. Consequently, the higher error rate can be found because the points are not properly selected. Thus, in this paper, an improved method is proposed to analyze for extracting a set of required points from the modification of the previous algorithms.

Algorithm

1) The inflection point has to be found by the change analysis from the sign of the slope in the following equation

\[
Slope = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}
\]

(24)

2) Inflection points are investigated to eliminate the rubbish points by the frequency of distance.

An example of rubbish point elimination is shown in Figure3. Inflection points in the original handwritten curve before and after the elimination of rubbish points are illustrated. It is noticed that after this process, some inflection points of the first steps will be eliminated.

3) To connect between two points, a straight line should be created to smooth the curve.

4) Two dimensional geometric transformations will be employed for translating a pivot point to the origin and rotating a given handwritten curve according to a straight line around x-axis

5) Keep maximum or minimum point of the curve.

6) Translate handwritten curve back to the original position.

7) For special cases, if we create a straight line between starting point and end point, then there may exists an intersection against the curve. This intersection point needs to be collected.

IV. RESULTS AND DISCUSSION

This section shows some examples in selecting a set of featured points including the results from approximating a handwritten curve. First, we will present the featured points of different handwritten curves obtained from above algorithm. Each curve varies according to the characteristics of the curves. Figure5 shows the examples of featured points that were collected from applying this algorithm to each curve. Started with a simple curve, the featured points for three curves, varying to the levels of curve complexity, are then derived.

After obtaining featured points, the next step is to approximate the curve by Bézier curve using algorithm in the previous work.

The first example is an exponential curve. It is illustrated since it is a simple curve. Three feature points were kept by algorithm can be seen in figure5 (a). The other examples are parabola and cusp curve. Both curves are more complicate than the exponential one. After applying our algorithm, 5 feature points can be accomplished as shown in figure5 (b,
c) and the last example is Sine wave curve. It is the most complex curve in this paper because 7 feature points were kept. Sine wave curve is a special case that the point participated in the intersection between straight line and curve in step 7 has to be kept. This situation can be depicted in figure 5 (d). Moreover, the results will be compared with the results from both algorithms in the previous work.

Fig. 6. The results of handwritten curve approximation by new algorithm (dot-dashed) compared to the results from previous algorithms: frequency (dash) and analyzing an inflection point (dot)

Fig. 7. The results of handwritten curves approximation by new algorithm (dot-dashed) compared to the results from previous algorithms: frequency (dash) and analyzing an inflection point (dot)

Fig. 8. The results of handwritten curves approximation by new algorithm (dot-dashed) compared to the results from previous algorithms: frequency (dash) and analyzing an inflection point (dot)

V. CONCLUSION

This paper presents a new algorithm for handwritten curve approximation by finding key-featured points. This is an improvement in reducing the deficiency of the previous algorithms. The results show that our new proposed algorithm can be eliminated disadvantages occurred in the existing methods. Furthermore, the results are closer to the given handwritten curves. However, some results in the examples are not satisfying because of the tardiness in the rotation of handwritten curve. Thus, in future work, the improvement should be devised for more efficient algorithm and better results.

REFERENCES