IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 16 May 2023, accepted 1 June 2023, date of publication 8 June 2023, date of current version 15 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3284128

RESEARCH ARTICLE

Adaptive Extension Fitting Scheme: An Effective Curve Approximation Method Using Piecewise Bézier Technology

XINGYU CUI[1], YONG LI[1], AND LILI XU[2]
[1]School of Statistics, Beijing Normal University, Haidian, Beijing 100875, China
[2]School of Applied Mathematics, Beijing Normal University, Zhuhai, Guangdong 519087, China

Corresponding author: Lili Xu (xulili@bnu.edu.cn)

ABSTRACT Curve approximation is a challenging issue to precisely depict exquisite shapes of natural phenomena, in which the piecewise Bézier curve is one of the most widely utilized tools due to its beneficial properties. It is essential to determine the quantity and location of control points through the process of generating the mathematical representation of desired objects. This paper presents a new algorithm called adaptive extension fitting scheme (AEFS) to determine a piecewise Bézier curve that best fits a given sequence of data points as well as locate the coordinates of the connecting points between the pieces adaptively. Taking full advantage of the scalability of the Bézier curve segment, AEFS is effective in sequential knot searching within an impressively small computational consumption. The capability of the proposed stepwise extension strategy is deduced from rigorous theoretical proof, resulting in proper connecting points together with well-fitted Bézier curves. The proposed algorithm is evaluated by some popular benchmarks for curve fitting, and compared with several state-of-the-art approaches. Experimental results indicate that AEFS outperforms other models involved in terms of execution time, fitting accuracy, number of segments, and the authenticity of shape contours.

INDEX TERMS Piecewise Bézier curve, Bézier curve segment, adaptive extension fitting, curve approximation, connecting point detection.

I. INTRODUCTION

Curve fitting is one of the essential tasks among applications in image processing, pattern recognition, computer graphics, and geophysics [1], [2], [3], [4]. Usually, the curve is intended to best indicate a real-world shape that otherwise has no mathematical representation or whose representation is unknown or too complicated. It requires determining a curve from a given sequence of points, which could be conducted by either interpolation or approximation. The interpolation methods determine a curve or function that goes through all the given points. Piegl and Tiller [5] created surfaces and reconstructed the object by determining these curves in computer-aided-design. Maekawa et al. [6] used the Newton-Raphson method to determine a B-spline curve for interpolation with high processing cost. Gofuku et al. [7] considered additional information to improve the geometric algorithm for plane curves, which was improved to the 3D sequence of points including the curvature vectors by Okaniwa et al. [8]. Although it may seem that interpolation methods accommodate curve fitting quite well, it should also be noted that real-world data always contain some measurement error which might cause wide curve variation even with the slightest change. The curve constructed based on a large number of data points would be too complicated to be editable with so many constraints.

Approximation, on the other way, determines a curve that is close to the given data in the best possible way without the obligation to pass exactly through all points. Bézier curve [9], [10] is one of the most widely utilized tools in the curve approximation process due to its beneficial geometric properties. Many approaches were proposed to

The associate editor coordinating the review of this manuscript and approving it for publication was Songwen Pei.

narrow down the discrepancy between a point and the approximating curve. Mineur et al. [11] proposed a fitting procedure controlled by the displacement of a given curve point for data fitting of the Bézier curve. In the literature [12], [13], the given sequence of points was approximated by one piece of the Bézier curve, though it is not practical in most scenarios. Basically, the classical Bézier curve is well-used in simple smooth curve approximation. A variety of aesthetic Bézier curves were proposed to improve the flexibility and adaptability, such as quintic trigonometric Bézier curve [14], general hybrid trigonometric Bézier curve [15], [16], and the generalized trigonometric Bézier curve [17] with shape parameters, respectively. In advance, Zain et al. [18] generalized a fractional Bézier curve for more flexibility and adjustability, yet the extra shape and fractional parameters would cause processing complexity.

The piecewise Bézier curve [19], [20], [21], [22], [23] is built as a union of Bézier curves connected end-to-end, whose characteristics are inherited to deal with more complex curve shapes. For instance, Yau and Wang [24] derived a piecewise cubic Bézier curve interpolating polynomial with the first order of continuity (C^1) while the method is time-consuming. Ueda et al. [25] used a new simulated annealing approach with the adaptive neighborhood to solve the curve fitting problem taking advantage of the local influence of each control point. Moreover, several approaches have been conducted to approximate the given sequence of points by the B-spline as the B-spline curve can be considered an extended version of the Bézier curve. Sobrinho et al. [2] determined a B-spline curve by the simulated annealing. Adi et al. [26] built the particle swarm optimization for NURBS which is commonly utilized in computer graphics [27]. Note that any piecewise Bézier curve with arbitrary degree can be converted into a B-spline and any B-spline can be converted into one or more Bézier curves [28]. However, the B-spline curve requires more computation and the underlying mathematics can be quite troublesome and intimidating at first.

The piecewise Bézier curve can be located and modified mainly by the control points, especially the connecting points between pieces of Bézier curve segments [15], [29]. The fitness of approximation of both Bézier curves and B-spline curves is highly relevant to their control points or knots. Therefore it is essential to determine the appropriate control points to approximate the sequence of given points. Generally, two reverse ideologies have been proposed and are widely used. One is expansive search [30], that is, keep integrating data points until the fit error exceeds a certain threshold after starting from the beginning. The other is the backward search through subdivision [20]. Converse to the expansive search, a single Bézier curve is adapted to fit all given data points at the first beginning. Once the error exceeds the pre-defined threshold, the data points will be divided into two subsets. The whole process stops until the data within every subset fit the required restriction. For easy operation, the bisection method [31], [32] is applied to find the knots at the subdivision step. Both methods are quite sensitive to local noisy data and badly defined tangents. In addition, Smith [33] presented a knot-selection strategy that equipped backward elimination procedures where the stepwise selection scheme would suffer from knot-compounding problems. Mao and Zhao [34] constructed a regression function through the non-linear least squares where the number of knots is determined by a modified generalized cross-validation device. Though the International Mathematical and Statistical Libraries routine [35] implemented in this article is convenient, fast and computationally stable, an unfortunate choice of initial knots may lead to drastically inappropriate local minima that would give misleading estimates. Sarfraz [36] presented a deterministic approach for approximating B-splines, including curvature-based knot detection, knot insertion, and a further knot-shifting procedure. It is costly to try all possible quantities and arrangements of control points to find the best fit.

Generally, the main methods of identifying control points of Bézier curves are using parametric optimization or searching, which requires a considerable amount of time and arithmetic resources. In view of the effectiveness of methodologies, we design an adaptive extension fitting scheme (AEFS) taking advantage of the piecewise Bézier curve. After explicitly distinguishing the definition of the Bézier curve and the Bézier curve segment, we raise the extension theorem exploring the capability of the Bézier curves, which provides strong theoretical support for the proposed fitting algorithm. With the guidance of the extension theorem, AEFS can select the connecting points adaptively avoiding time consumption caused by aimless search. Meanwhile, the quantity of control points is automatically determined without initial value so that the selection of knots is not influenced by different initialization. In this way, the proposed method can innovatively depict the curve trend under a suitable prescribe error bound, resulting in well-approximated piecewise Bézier curve. Experiments are also conducted to verify the performance of our proposed algorithm.

The rest of the paper is organized as follows: Section II clarifies the definition of the Bézier curve and the Bézier curve segment whose properties are utilized for the extension theorem. In Section III, we outline the adaptive extension fitting scheme and detail its key techniques. Section IV shows the experimental performances of our algorithm together with other compared methods on multiple datasets. Finally, the conclusion and discussion are drawn in Section V.

II. DEFINITIONS AND PROPERTIES

The bi-dimensional curve approximation problem focuses on obtaining a curve with explicit expressions that approach the given sequential points. One piece or piecewise Bézier curves are one of the most commonly used candidates owing to their preferably geometrical properties. In this section, some concepts and properties related to the Bézier curve are clarified so as to cosily draw into our method.

A. Bézier CURVE AND Bézier CURVE SEGMENT

Referring to the introductory book [1], every polynomial curve $B(t)$ of degree $\leq n$ has a unique nth degree Bézier representation

$$B(t) = \sum_{i=0}^{n} \boldsymbol{x}_i B_i^n(t). \tag{1}$$

where the coefficients $\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathbb{R}^d$, and $B_i^n(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}$ are the classical Bernstein polynomials of degree n.

Here, for clarity of presentation, we introduce the strict mathematical definition of Bézier curves. An nth degree **Bézier curve** is denoted as follows,

$$\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)} = \left\{ B(t) \in \mathbb{R}^d \,| t \in \mathbb{R} \right\}. \tag{2}$$

Subsequently, a segment of the nth degree Bézier curve $\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}$ over interval $[a, b]$ is

$$\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}[a, b] = \left\{ B(t) \in \mathbb{R}^d \,| t \in [a, b]\,,\ a < b \right\}, \tag{3}$$

which is briefly noted **Bézier curve segment**.

B. STANDARD REPRESENTATION OF Bézier CURVE SEGMENT

For any Bézier curve segment $\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}[a, b]$, an affine parameter transformation $t(u) = a(1-u) + bu,\ u \in [0, 1]$ does not change its degree and shape. Consequently, there exists a point sequence $\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_1, \cdots, \tilde{\boldsymbol{x}}_n \in \mathbb{R}^d$ such that

$$\tilde{B}(u) \triangleq \sum_{i=0}^{n} \tilde{\boldsymbol{x}}_i B_i^n(u),\ u \in [0, 1]. \tag{4}$$

satisfying $\tilde{\boldsymbol{x}}_0 = B(a), \tilde{\boldsymbol{x}}_n = B(b)$. The points $\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_1, \cdots, \tilde{\boldsymbol{x}}_n$ are **Bézier points** or **control points**, while Eq.(4) is the **standard representation** of $\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}[a, b]$. In particular, $B(t) = \sum_{i=0}^{n} \boldsymbol{x}_i B_i^n(t),\ t \in [0, 1]$ is the standard representation of $\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}[0, 1]$.

The existence of the standard Bézier representation is straightforward, as is its uniqueness. We can easily obtain the following Lemma 1 by combining the mentioned affine parameter transformation with the conversion between the Bernstein polynomials and power basis functions.

Lemma 1: For any Bézier curve segment $\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots\,\boldsymbol{x}_n)}[a, b]$, its standard representation exists and is uniquely defined.

In practical implementation, it is not adequate to merely know the existence and uniqueness, but also the specific expressions, which can be illustrated in Lemma 2. [37]

Lemma 2: For any Bézier curve segment $\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots\,\boldsymbol{x}_n)}[a, b]$, it has control points satisfying

$$\begin{bmatrix} \tilde{\boldsymbol{x}}_0 \\ \tilde{\boldsymbol{x}}_1 \\ \vdots \\ \tilde{\boldsymbol{x}}_n \end{bmatrix}_{(n+1)\times d} = \rho_n \begin{pmatrix} b-a & 0 \\ a & 1 \end{pmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix}_{(n+1)\times d}, \tag{5}$$

where $\rho_n(\cdot)$ is a homomorphism. Specifically, $\rho_n\left(A_{a,b}\right) = \delta_n\left(M_1^{-1} A_{a,b} M_1\right)$, where $M_1 = \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix}$, $\delta_n\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \Psi^{-1}\begin{pmatrix} (a+bx)^n \\ (a+bx)^{n-1}(c+dx) \\ (a+bx)^{n-2}(c+dx)^2 \\ \vdots \\ (c+dx)^n \end{pmatrix}$, and $\Psi\left(M_{(n+1)\times(n+1)}\right)$ is the column vector of polynomials obtained by applying ψ to every row, $\psi(a_0, a_1, \cdots, a_n) = a_0 + a_1 x + \cdots + a_n x^n$.

C. EXTENSION THEOREM

According to the connotations clarified above, it is essential to exploit the information comprised in a Bézier curve segment for curve approximation. In this case, we can prove that any Bézier curve segment could hold all the information of its originated Bézier curve as shown in Theorem 1.

Theorem 1: For any Bézier curve segment $\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}[a, b]$, its standard representation can uniquely expand to the Bézier curve it comes from.

In details, let $\tilde{B}(u) = \sum_{i=0}^{n} \tilde{\boldsymbol{x}}_i B_i^n(u), u \in [0, 1]$ be the standard representation of $\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}[a, b], \forall\ a < b$, where $\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_1, \cdots, \tilde{\boldsymbol{x}}_n$ are the known control points. Then, the expanded Bézier curve $\tilde{\mathcal{B}}_{(\tilde{\boldsymbol{x}}_0,\tilde{\boldsymbol{x}}_1,\cdots,\tilde{\boldsymbol{x}}_n)} = \left\{ \tilde{B}(u) \in \mathbb{R}^d \,| u \in \mathbb{R} \right\}$ is exactly the same as the Bézier curve $\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}$, that is $\tilde{\mathcal{B}}_{(\tilde{\boldsymbol{x}}_0,\tilde{\boldsymbol{x}}_1,\cdots,\tilde{\boldsymbol{x}}_n)} = \mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}$.

Proof: The proof may differ slightly due to the size relationship of $\{0, 1, a, b\}$. Here, we start considering the situation that $1 < a < b$. For simplicity of presentation, we denote $X^T = \left[\boldsymbol{x}_0^T\ \boldsymbol{x}_1^T\ \cdots \boldsymbol{x}_n^T\right]$ and $\tilde{X}^T = \left[\tilde{\boldsymbol{x}}_0^T\ \tilde{\boldsymbol{x}}_1^T\ \cdots \tilde{\boldsymbol{x}}_n^T\right]$. Then following Lemma 1 and Lemma 2, we can quickly built a relationship between two point sequences, $\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ and $\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_1, \cdots, \tilde{\boldsymbol{x}}_n$, that is

$$\tilde{X} = \rho_n \begin{pmatrix} b-a & 0 \\ a & 1 \end{pmatrix} X. \tag{6}$$

For $\forall\ B(d) \in \mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}$, if $d \in [a, b]$, then $B(d) \in \mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}[a, b] = \tilde{\mathcal{B}}_{(\tilde{\boldsymbol{x}}_0,\tilde{\boldsymbol{x}}_1,\cdots,\tilde{\boldsymbol{x}}_n)}[0, 1] \subset \tilde{\mathcal{B}}_{(\tilde{\boldsymbol{x}}_0,\tilde{\boldsymbol{x}}_1,\cdots,\tilde{\boldsymbol{x}}_n)}$. If $d > b$, using Lemma 2, the control points of $\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}[a, d]$ satisfy $\rho_n \begin{pmatrix} d-a & 0 \\ a & 1 \end{pmatrix} X$. Considering Bézier curve segment $\tilde{\mathcal{B}}_{(\tilde{\boldsymbol{x}}_0,\tilde{\boldsymbol{x}}_1,\cdots,\tilde{\boldsymbol{x}}_n)}\left[0, \frac{d-a}{b-a}\right]$, its control points follows

$$\begin{aligned} \rho_n \begin{pmatrix} \frac{d-a}{b-a} & 0 \\ 0 & 1 \end{pmatrix} \tilde{X} &= \rho_n \begin{pmatrix} \frac{d-a}{b-a} & 0 \\ 0 & 1 \end{pmatrix} \rho_n \begin{pmatrix} b-a & 0 \\ a & 1 \end{pmatrix} X \\ &= \rho_n \begin{pmatrix} d-a & 0 \\ a & 1 \end{pmatrix} X. \end{aligned} \tag{7}$$

Thus, $B(d) \in \mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}[a, d] = \tilde{\mathcal{B}}_{(\tilde{\boldsymbol{x}}_0,\tilde{\boldsymbol{x}}_1,\cdots,\tilde{\boldsymbol{x}}_n)}\left[0, \frac{d-a}{b-a}\right] \subset \tilde{\mathcal{B}}_{(\tilde{\boldsymbol{x}}_0,\tilde{\boldsymbol{x}}_1,\cdots,\tilde{\boldsymbol{x}}_n)}$. We can prove $B(d) \in \tilde{\mathcal{B}}_{(\tilde{\boldsymbol{x}}_0,\tilde{\boldsymbol{x}}_1,\cdots,\tilde{\boldsymbol{x}}_n)}$ in the same way when $d < a$.

On the other hand, we can prove $\tilde{B}(z) \in \mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}$ for $\forall\ \tilde{B}(z) \in \tilde{\mathcal{B}}_{(\tilde{\boldsymbol{x}}_0,\tilde{\boldsymbol{x}}_1,\cdots,\tilde{\boldsymbol{x}}_n)}$ in a similar process. Thus,

$\tilde{\mathcal{B}}_{(\tilde{x}_0,\tilde{x}_1,\cdots,\tilde{x}_n)} = \mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}$ holds for $1 < a < b$. And the rest situations can be proven similarly.

So far, we have proved that $\tilde{\mathcal{B}}_{(\tilde{x}_0,\tilde{x}_1,\cdots,\tilde{x}_n)} = \mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}$ for all situations of $\{0, 1, a, b\}$. To sum up, we can draw the conclusion that the Bézier curve generated by the standard representation of $\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}[a, b]$ is the same as $\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}$. □

Theorem 1 is named as the extension theorem due to its geometrical extension. Following this theorem, any Bézier curve segment with the standard representation $B(t) = \sum_{i=0}^{n} \boldsymbol{x}_i B_i^n(t), t \in [0, 1]$ can uniquely extend to Bézier curve $\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}$. Here we just focus on the extended Bézier curve segment rather than the whole Bézier curve. Corollary 1 illustrates the curve segment extending in a single parametric direction to a wilder Bézier curve segment together with its standard representation while Corollary 2 presents the case of two parametric directions accordingly.

Corollary 1: If a Bézier curve segment $B(t) = \sum_{i=0}^{n} \boldsymbol{x}_i B_i^n(t), t \in [0, 1]$ *extends from* $[0, 1]$ *in larger parametric direction to* $[0, d]$, $d > 1$, *then* $B(t) = \sum_{i=0}^{n} \boldsymbol{x}_i B_i^n(t), t \in [0, d]$ *is also a Bézier curve segment, and its control points* $\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_1, \cdots, \tilde{\boldsymbol{x}}_n$ *satisfying*

$$\begin{bmatrix} \tilde{\boldsymbol{x}}_0 \\ \tilde{\boldsymbol{x}}_1 \\ \vdots \\ \tilde{\boldsymbol{x}}_n \end{bmatrix}_{(n+1)\times d} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1-d & d & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ (1-d)^n & n(1-d)^{n-1}d & \cdots & d^n \end{pmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix}_{(n+1)\times d}. \quad (8)$$

Corollary 2: If a Bézier curve segment $B(t) = \sum_{i=0}^{n} \boldsymbol{x}_i B_i^n(t), t \in [0, 1]$ *extends from* $[0, 1]$ *in two parametric directions to* $[a, b]$, *where* $a < 0 < 1 < b$. *Then* $B(t) = \sum_{i=0}^{n} \boldsymbol{x}_i B_i^n(t), t \in [a, b]$ *is a Bézier curve segment too, and its control points* $\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_1, \cdots, \tilde{\boldsymbol{x}}_n$ *satisfying*

$$\begin{bmatrix} \tilde{\boldsymbol{x}}_0 \\ \tilde{\boldsymbol{x}}_1 \\ \vdots \\ \tilde{\boldsymbol{x}}_n \end{bmatrix}_{(n+1)\times d} = \Psi^{-1} \begin{pmatrix} [(1-a)+ax]^n \\ [(1-a)+ax]^{n-1}[(1-b)+bx] \\ [(1-a)+ax]^{n-2}[(1-b)+bx]^2 \\ \vdots \\ [(1-b)+bx]^n \end{pmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix}_{(n+1)\times d}, \quad (9)$$

where Ψ *is the same as in Lemma 2.*

With the assistance of this fact, any Bézier curve segment can generate any part of the Bézier curve it represents. In curve approximation, the representative range of a great majority of fitted Bézier curve segments should not be limited to the involved data points, but an expanded range. Thus, a more suitable Bézier curve segment can be computed immediately by extending a locally well-approximated Bézier segment curve as far as possible rather than fitting over and over again. Fully taking the scalability of the Bézier curve segment into consideration, we can depict data points in a better-fitted curve.

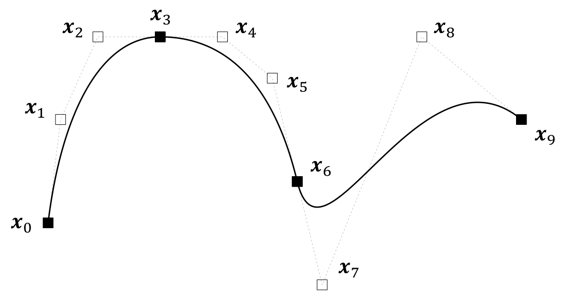

FIGURE 1. **Piecewise cubic Bézier curve with 3 Bézier curve segments. The control points are represented by squares, especially in which x_3 and x_6 are two connecting points. Note that control points x_2, x_3 and x_4 are colinear, so are x_5, x_6 and x_7.**

Sometimes, the degree of Bézier curves needs to be continuously increased to adapt to the changeable data points. Nevertheless, for the entire Bézier curve, it is inevitable to suffer the consumption from modifying even one control point. In practical application, a sequence of Bézier curve segments with a low degree usually shows better performances than one Bézier curve with a high degree.

D. PIECEWISE Bézier CURVES

Consisting of a sequence of Bézier curves, the piecewise Bézier curve can smartly avoid the locally uncontrollable, where only partial curves are affected by the modified control point. In terms of its function, the connecting point overlaps both the last control point of the previous Bézier curve segment and the first control point of the adjacent next Bézier curve segment. Meanwhile, the control points adjacent to a connecting control point need to satisfy the following relationship

$$\boldsymbol{x}_{n(i+1)-1} = \boldsymbol{x}_{n(i+1)} - (\boldsymbol{x}_{n(i+1)+1} - \boldsymbol{x}_{n(i+1)-1})\beta_{i+1}, \quad (10)$$

for $i = 0, 1, \cdots, m$, where n is the degree of each Bézier curve segment, m is the number of curve segments and β_{i+1} is the continuity factor.

As one significant trait, Bézier curve $\mathcal{B}_{(\boldsymbol{x}_0,\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n)}$ passes through $\boldsymbol{x}_0$ when $t = 0$ and through $\boldsymbol{x}_n$ when $t = 1$, and it is tangent at $\boldsymbol{x}_0$ to the vector from $\boldsymbol{x}_0$ to $\boldsymbol{x}_1$ and tangent at $\boldsymbol{x}_n$ to the vector from $\boldsymbol{x}_{n-1}$ to $\boldsymbol{x}_n$. Hence, the above conditions impose constraints on any two adjacent Bézier curve segments, promising that they are connected and their tangent lines at the connecting point coincide. Furthermore, the smoothness at the connecting point can be enhanced by imposing more constraints. Fig. 1 shows an example of a piecewise Bézier curve, which consist of three cubic Bézier curve segments.

In the next section, we would formally propose the adaptive extension fitting scheme utilizing piecewise Bézier curves with the guidance of the extension theorem.

III. ADAPTIVE EXTENSION FITTING SCHEME

In this section, we construct an adaptive algorithm that can detect the connecting points without tough calculations based on the extension theorem. The general process is described first, after which the initialization strategy for the first curve and the Bézier curve extension technology are detailed in turn. Finally, an adjusted initialization strategy for the second and subsequent curves is elaborated, so is the full algorithm.

A. OVERVIEW

The insight is dedicated to the determination of suitable positions for connecting points, which is one of the most essential tasks of piecewise Bézier curve approximation. Well-determined connecting points lead to superb performance with few segments, acceptable discrepancy errors, and less time consumption. In general, researchers are used to grid search for optimal connecting points or regard connecting points as parameters for optimization, both of which suffer computation consumption. The quantity of control points is usually unknown yet hard to decide. To alleviate this hassle, we propose the adaptive extension fitting scheme (AEFS) to effectively detect the connecting points in the fitting of piecewise Bézier curves, taking advantage of its unique extension property. Two main stages are constructed. We first approximate a small segment of data points to an initial Bézier curve. Generally the fitted Bézier curve segment is not assessed to be the best fit, incurring its endpoints are neither the appropriate connecting points, we enlarge the scope of the initial Bézier curve segment with the assistance of Corollary 1 (or Corollary2) of Theorem 1 immediately afterward. That is, the first piece of curve is gained by extending the initial Bézier curve segment until the optimal connecting point is identified satisfying the assessment criteria, which balances the length of the current curve segment with the flexibility of the subsequent. Once the connecting point is determined, the standard representation of the extended curve segment can be computed instantly.

So far, we accomplish the curve fitting of one piece Bézier curve. The approximation of the subsequent pieces is similar, apart from that condition (10) as an additional constraint is imposed on the subsequent curve fitting. Thus, a sequence of suitable Bézier curves is fitted progressively. Fig.2 summarizes the steps for the proposed adaptive extension fitting scheme.

In the following, we consider utilizing the second-degree Bézier curve to detail the key techniques of the proposed algorithm.

B. INITIALIZATION STRATEGY FOR FIRST CURVE

Under a prescribe bound, the curve initialization strategy can be broadly divided into two parts: fitting an initial Bézier curve $\boldsymbol{p}(t)$ and testing the discrepancy between $\boldsymbol{p}(t)$ and data points.

1) FIT AN INITIAL Bézier CURVE $p(t)$

In the context of Bézier approximation, the constrained least-squares method is a common way to approximate a Bézier curve, which can be regarded as the initial Bézier curve. Usually, the fitted curve is requested to pass through the endpoints of the involved data points.

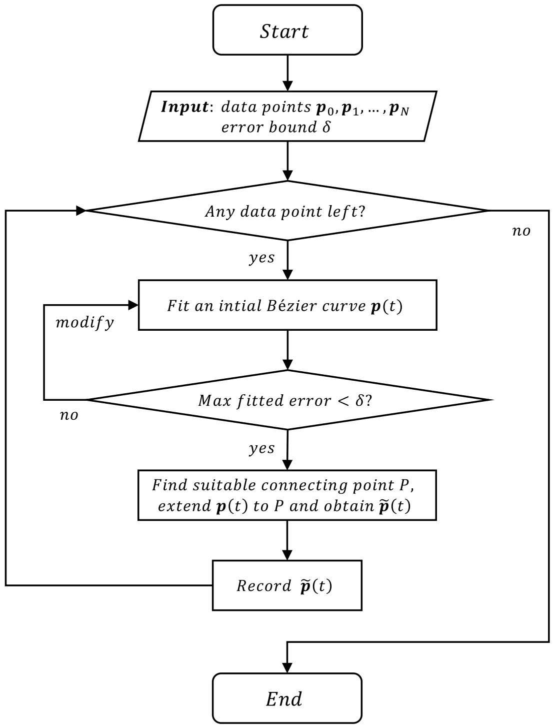

FIGURE 2. The flowchart of adaptive extension fitting scheme.

Using the constrained least-squares method [1], a sequence of data points $\boldsymbol{p}_0, \boldsymbol{p}_1, \cdots, \boldsymbol{p}_S$ can be depicted by a Bézier curve, where $\boldsymbol{p}_s = (p_{s1}, p_{s2}) \in \mathbb{R}^2$ $(s = 0, 1, \cdots, S)$. Here, we suppose the initial curve holds the standard quadratic Bézier representation $\boldsymbol{p}(t) = \boldsymbol{x}_0 B_0^2(t) + \boldsymbol{x}_1 B_1^2(t) + \boldsymbol{x}_2 B_2^2(t)$, $t \in [0, 1]$ following Eq. (4) with $n = 2$, in which $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2$ are the control points to be estimated, and $\boldsymbol{x}_i = (x_{i1}, x_{i2}) \in \mathbb{R}^2$, $i = 0, 1, 2$. Once the parameters $t_0, t_1, \cdots, t_S$ are determined, also marked as $\boldsymbol{t} = (t_0, t_1, \cdots, t_S)$, we only need to minimize the difference between the estimated points $\boldsymbol{p}(t_0), \boldsymbol{p}(t_1), \cdots, \boldsymbol{p}(t_S)$ and the corresponding data points. Especially, the endpoints of fitted curve are limited to point $\boldsymbol{p}_0$ and point $\boldsymbol{p}_S$. Thus, the system to be solved is

$$\left[\begin{array}{c|c} \widetilde{\boldsymbol{B}}^T\widetilde{\boldsymbol{B}} & \boldsymbol{D}^T \\ \hline \boldsymbol{D} & \boldsymbol{O} \end{array}\right] \left[\begin{array}{c} \boldsymbol{x} \\ \hline \boldsymbol{y} \end{array}\right] = \left[\begin{array}{c} \widetilde{\boldsymbol{B}}^T\boldsymbol{p} \\ \hline \boldsymbol{q} \end{array}\right]. \tag{11}$$

where $\boldsymbol{O}$ is zero matrix,

$$\widetilde{\boldsymbol{B}} = \widetilde{\boldsymbol{B}}(\boldsymbol{t}) = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{B} \end{bmatrix}, \tag{12}$$

$$\boldsymbol{B} = \begin{bmatrix} B_0^2(t_0) & B_1^2(t_0) & B_2^2(t_0) \\ B_0^2(t_1) & B_1^2(t_1) & B_2^2(t_1) \\ \vdots & \vdots & \vdots \\ B_0^2(t_S) & B_1^2(t_S) & B_2^2(t_S) \end{bmatrix}, \tag{13}$$

$$\boldsymbol{D} = \begin{bmatrix} 1\ 0\ 0\ 0\ 0\ 0 \\ 0\ 0\ 1\ 0\ 0\ 0 \\ 0\ 0\ 0\ 1\ 0\ 0 \\ 0\ 0\ 0\ 0\ 0\ 1 \end{bmatrix}, \tag{14}$$

and $\boldsymbol{x} = [x_{01}\ x_{11}\ x_{21}\ x_{02}\ x_{12}\ x_{22}]^T$ is to be estimated, $\boldsymbol{p} = [p_{01}, p_{11}, \cdots, p_{S1}, p_{02}, p_{12}, \cdots, p_{S2}]^T$, $\boldsymbol{q} = [p_{01}\ p_{S1}\ p_{02}\ p_{S2}]^T$. Especially, $\boldsymbol{D}$ is the constrained matrix for the first Bézier curve, and the restriction can be expressed as $\boldsymbol{Dx} = \boldsymbol{q}$.

In addition, a number of methods can be chosen for the parameterization, including the uniformly spaced method, chord-length method, centripetal method and other iterative methods. In this paper, we integrate the chord-length method [27] into the curve fitting to initialize parameters $t_0, t_1, \cdots, t_S$ when the parameters are unknown.

2) DISCREPANCY TEST

After the initial Bézier approximation, we should test the fitting goodness of the Bézier curve $\boldsymbol{p}(t)$, $t \in [0, 1]$ according to the distance from the involved data points to the curve. The ideal way to calculate the distance from any point P to the approximated Bézier curve segment $\boldsymbol{p}(t)$ is the perpendicular distance by projecting the point to the curve. For a parametric equation, we need to determine the parameter t_0 corresponding to the point P at the beginning. A simple approximation is to minimize the distance, i.e.

$$\min_t |\boldsymbol{p}(t) - P|^2 . \tag{15}$$

In this scenario, the parameter t is searched on [0, 1] because the point to be calculated has participated in the curve fitting.

Once the parameter t_0 is known, the distance can be easily calculated by $|\boldsymbol{p}(t_0) - P|$, also named discrepancy. And the fitting discrepancy is the sum of the discrepancies over all the participated data points, which is an important index to depict the fitting goodness. Since the number of involved data points is very important to the initial approximated curve, once the fitting is poor, we can simply reduce it to improve the fitting goodness.

Combining the above two parts, the pseudocode of the whole initialization strategy is shown in Algorithm 1. Given the key technique, we named it the constrained least-squares method.

C. CURVE EXTENSION

Up to now, we have obtained an initial Bézier curve within a tolerable fitting error. For fear that additional time consumption might be caused by non-optimal locations of control points, the determination of the stopping position can be achieved in one step through curve extension since the representativeness of the fitted Bézier curve segment is not only confined to the involved data points.

Fig. 3 illustrates the curve extension of a quadratic Bézier curve, where the Bézier curve segment $\mathcal{B}_{(x_0,x_1,x_2)}[0, 1]$ with its standard representation $B(t) = \boldsymbol{x}_0 B_0^2(t) + \boldsymbol{x}_1 B_1^2(t) + \boldsymbol{x}_2 B_2^2(t)$, $t \in [0, 1]$ extends to $t \in [0, d]$, $d > 1$. Here, the

Algorithm 1 Constrained Least-Squares Method

Input: Sequence of points $\boldsymbol{p}_0, \boldsymbol{p}_1, \cdots, \boldsymbol{p}_S$, marked as $\boldsymbol{p_{seq}}$; the error threshold δ; the constrained matrix $\boldsymbol{D}$.
Output: Control points $(\boldsymbol{x_0}, \boldsymbol{x_1}, \boldsymbol{x_2})$; parameter $\boldsymbol{t}$.
procedure CONSTRAINEDLS ($\boldsymbol{p_{seq}}, \boldsymbol{D}, \delta$)
  Initialize parameter $\boldsymbol{t}$, discrepancy $\boldsymbol{d} = (\infty, \cdots, \infty)$
  while $\max(\boldsymbol{d}) > \delta$
    Calculate design matrix $\widetilde{\boldsymbol{B}}(\boldsymbol{t})$ using Eq.(12)(13)
    Find minimal solution $\boldsymbol{x}, \boldsymbol{y}$ of the linear system (11)
    Update $\boldsymbol{t}$ pointwise through Eq.(15)
    Calculate pointwise discrepancy $\boldsymbol{d}[s] = |\boldsymbol{p}(\boldsymbol{t}[s]) - \boldsymbol{p}_s|$
    Record $\tilde{S} = S, \tilde{\boldsymbol{t}} = \boldsymbol{t}$
    Update $S = S - 1, \boldsymbol{t} = \boldsymbol{t}[1:S]/\boldsymbol{t}[S]$
  end while
  return $(\boldsymbol{x_0}, \boldsymbol{x_1}, \boldsymbol{x_2})$, $\tilde{S}$, $\tilde{\boldsymbol{t}}$
end procedure

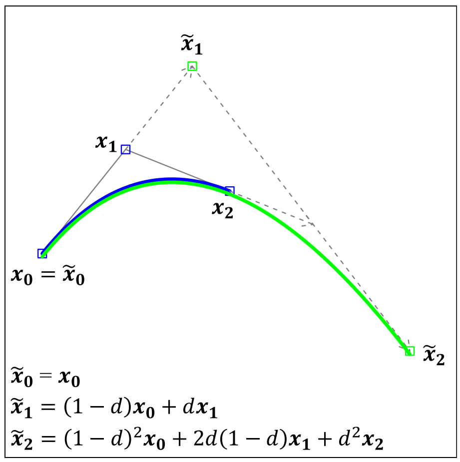

FIGURE 3. Curve extension of a quadratic Bézier curve. The blue curve x_0x_2 extends to the green curve $\tilde{x}_0\tilde{x}_2$, and the squares are the control points indicated in corresponding color.

Bézier points of the extended curve are $\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_1$ and $\tilde{\boldsymbol{x}}_2$ obtained by using Corollary 1.

To determine the ideal position of the connecting point, we first test the closeness from the subsequent uninvolved data points to the initial Bézier curve $\boldsymbol{p}(t)$ on the basis of Eq. (15), where $[1, \infty)$ is the search region of parameter t. If the unfitted points along the sequential direction are close to the initial Bézier curve $\boldsymbol{p}(t)$, we can extend the approximated curve segment to a point $\boldsymbol{p_I}$ with parameter $t = d > 1$. Then we can obtain the standard representation of the extended Bézier curve segment by referring to Corollary 1 to ensure the simplicity of the curve representation. Note that the parametric normalization only modifies its expression and keeps the curve shape unchanged. In detail, if $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2$ are the initial control points, the control points of the extended curve are $\boldsymbol{x}_0$, $d(\boldsymbol{x}_1 - \boldsymbol{x}_0) + \boldsymbol{x}_0$, $d^2(\boldsymbol{x}_0 - 2\boldsymbol{x}_1 + \boldsymbol{x}_2) + 2d(\boldsymbol{x}_1 - \boldsymbol{x}_0) + \boldsymbol{x}_0$, where d is the corresponding parameter towards $\boldsymbol{p_I}$. The pseudocode of the curve extension is presented in Algorithm 2.

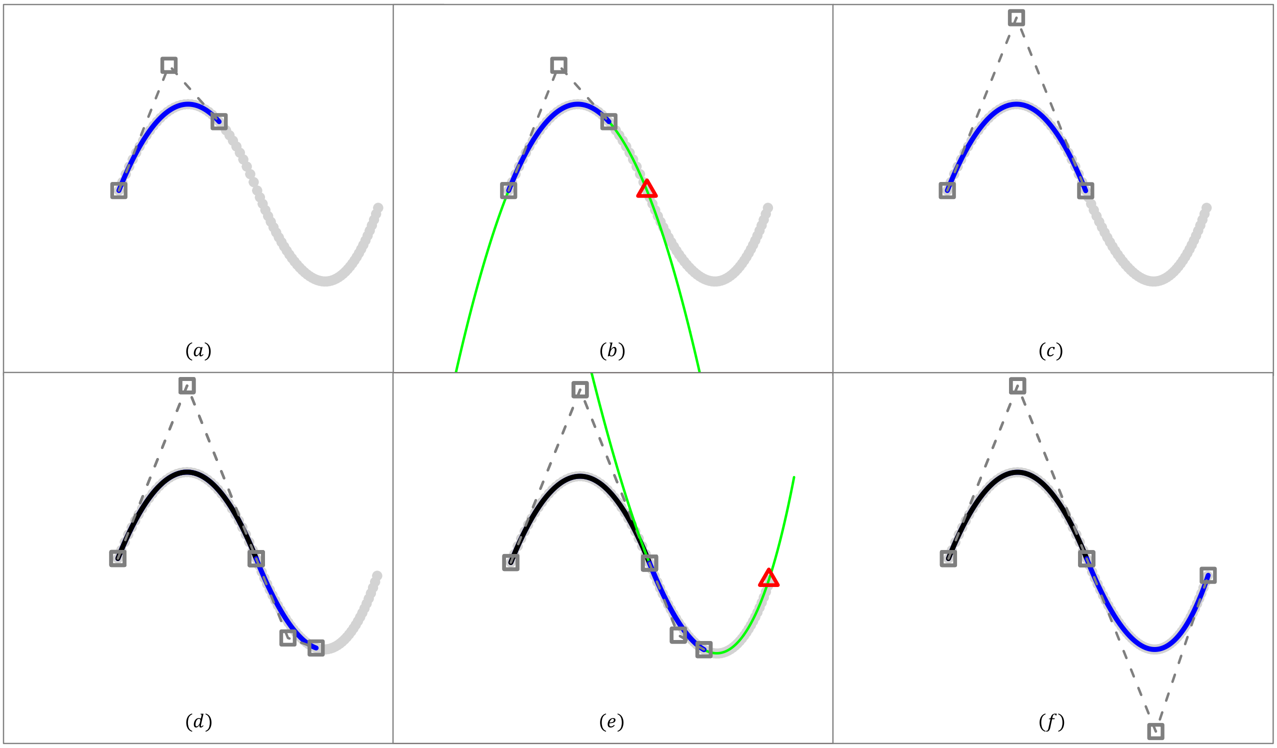

FIGURE 4. The illustration of curve approximation utilizing AEFS.

Algorithm 2 Curve Extension

Input: Sequence of uninvolved points $p_{\tilde{S}+1}, p_{\tilde{S}+2}, \cdots, p_N$, marked as p_{seq}; the control points of initial fitted curve (x_0, x_1, x_2), marked as x; the error threshold δ.

Output: Control points $(\tilde{x}_0, \tilde{x}_1, \tilde{x}_2)$.

procedure CURVE EXTENSION (p_{seq}, x, δ)
  dis $\leftarrow$ Calculate the distance from p_{seq} to
    $p(t) = x_0\,(1-t)^2 + x_1\,2t(1-t) + x_2\,t^2, t > 1$
  $I, d \leftarrow$ Compare ***dis*** with δ to determine the stopping data point index I of p_{seq} and its parameter d
  Calculate
    $\tilde{x}_0 = x_0$
    $\tilde{x}_1 = d(x_1 - x_0) + x_0$
    $\tilde{x}_2 = d^2(x_0 - 2x_1 + x_2) + 2d(x_1 - x_0) + x_0$
  return $(\tilde{x}_0, \tilde{x}_1, \tilde{x}_2), I$
end procedure

D. SUBSEQUENT PROCESSING

So far, the approximation procedure for the first piece Bézier curve is completed. The proposed strategy makes full use of the properties of the Bézier curve which can approximate the Bézier curve effectively and efficiently. If the fitted curve can cover all the data points, the curve fitting ends. If there still remains data points, we can easily obtain the further approximated Bézier curves smoothly by repeating this procedure.

One thing to be noted, for the fitting of the second and subsequent curves, is that the new constrained equations $\widetilde{D}x = \tilde{q}$ add an additional condition (10) to keep C^1 continuity with the preceding approximated curve. Specifically, we marked the last fitted curve as $p^{k-1}(t)$, and its control points are represented by $(x_0^{k-1}, x_1^{k-1}, x_2^{k-1})$. And the index of the last involved data point is I^{k-1}. Then for the curve fitting of data points $p_{I^{k-1}}, p_{I^{k-1}+1}, \cdots, p_{I^{k-1}+S}$, the detailed constrained equations are as follows,

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & x_{22}^{k-1} - x_{12}^{k-1} & 0 & 0 & x_{11}^{k-1} - x_{21}^{k-1} & 0 \end{bmatrix} \begin{bmatrix} x_{01}^k \\ x_{11}^k \\ x_{21}^k \\ x_{02}^k \\ x_{12}^k \\ x_{22}^k \end{bmatrix} = \begin{bmatrix} x_{21}^{k-1} \\ p_{I^{k-1}+S,1} \\ x_{22}^{k-1} \\ p_{I^{k-1}+S,2} \\ x_{11}^{k-1}(x_{22}^{k-1} - x_{12}^{k-1}) - x_{12}^{k-1}(x_{21}^{k-1} - x_{11}^{k-1}) \end{bmatrix}. \tag{16}$$

Generally, AEFS is presented as an adaptive fitting method using piecewise Bézier technology. Firstly, an initial Bézier curve is obtained with the fitting accuracy restricted by a prescribe bound. Meanwhile, the size of the involved data points is adjusted to control the fitting error. The Algorithm 1 details the initialization strategy. Then the obtained curve segment is extended to its best stopping position while the standard representation of the first fitted Bézier curve segment is calculated according to Algorithm 2.

Note that the last data point involved in the previous fitted curve has become the start of the immediately following fitted curve. Through continuously repeating the initialization and curve extension, we could finally obtain a sequence of well-fitted Bézier curve segments eliminating unnecessary computational consumption. In summary, we brief the complete procedure of adaptive extension fitting scheme with the pseudocode shown in Algorithm 3.

Algorithm 3 Adaptive Extension Fitting Scheme

Input: Sequence of data points $\boldsymbol{p}_0, \boldsymbol{p}_1, \cdots, \boldsymbol{p}_N$, marked as $\boldsymbol{p_{seq}}$; the error threshold δ.

Output: Control points sequence $\left(\tilde{x}_0^k, \tilde{x}_1^k, \tilde{x}_2^k\right)$, $k = 1, 2, \cdots, m$.

```
procedure AEFS (p_seq, δ)
  Initialize l = 0, r = min{10, N}, k = 1
  while l < N
    if k = 1
      (x_0^k, x_1^k, x_2^k), S̃ ← CONSTRAINEDLS (p_seq[l : r], D, δ)
    else
      (x_0^k, x_1^k, x_2^k), S̃ ← CONSTRAINEDLS (p_seq[l : r], D̃, δ)
    Update r = l + S̃, and record x = (x_0^k, x_1^k, x_2^k)
    (x̃_0^k, x̃_1^k, x̃_2^k), I ← CURVE EXTENSION (p_seq[r : N], x, δ)
    Update r = l + S̃ + I
    return (x̃_0^k, x̃_1^k, x̃_2)^k, r
    Update l = r, r = min{l + 10, N}, k = k + 1
  end while
end procedure
```

For visual clarity, Fig.4 illustrates the details of the whole curve fitting process step by step, producing two smoothly connected fitted curves. Fig. 4 (a) illustrates the fitted initial Bézier curve segment of the first curve piece whose control points are represented by three hollow squares. Then Fig. 4 (b) illustrates the curve extension for the initial Bézier curve of the first curve piece, where the stopping point is represented by a red triangle. In the curve approximation process, we tentatively choose the maximum extension point and go back several points as the final stopping position under the prescribed error bound. Here, we successfully obtain the first Bézier curve piece of the data sequence, as illustrated in Fig. 4 (c). After that, we should estimate the second curve piece on the residual data point, starting from the end point of the previous fitted curve. Fig. 4 (d) illustrates the fitted initial Bézier curve segment of the second curve piece. Besides the constrains on the endpoints, we can easily see that the approximation procedure obeys the smooth connecting rule. And following the same procedure, Fig. 4 (e) shows the process that the initial curve segment tries to stop at the best position, and Fig. 4 (f) subsequently shows the second fitted Bézier curve piece appended to the first one.

IV. EXPERIMENTS

The performance of our adaptive extension fitting scheme (AEFS) is implemented and compared in this section. Our algorithm was tested on three distinct examples, wherein the first inflection curve is a simple case connected by two pieces of Bézier curves with C^1 continuity. The second data set distributes like a butterfly, which describes a complex curve generated from deterministic functions with a lot of self-intersections, and the third test is carried out on a set of curve-based strokes of seal characters.

A. BRIEF INTRODUCTION OF METHODS IN COMPARISON

A total of six state-of-the-art models were selected for the experimental phase besides AEFS. For the convenience of the following analysis and description, all mentioned comparative methods are named with their first author for better illustration. The structural details are described below.

1) **Smith** [33] presented a knot-selection strategy that equipped backward elimination procedures to fit splines using statistical variable selection techniques. A pool of knots is fixed in an equidistant way, and because of the computational overheads for each fit, the number of knots is initialized as one-fifth of the data points for comparison.
2) **MaoZhao** [34] regarded the knot locations and their number as variables and constructed a regression function through non-linear least squares, where the number of knots is determined by a modified generalized cross-validation (GCV) device. This method also rely on several statistical techniques. For easy comparison, the initial number of knots is varied around our method while the initial knot positions are set in an equidistant way.
3) **Sarfraz** [36] presented a deterministic approach for approximating B-splines, including curvature-based knot detection, knot insertion, and a further knot shifting procedure. We take turns the window size from 1 to 10 and choose 2 for the curvature measure as the experimental setting.
4) **Ueda** [25] designed a metaheuristic algorithm using a simulated annealing scheme with the adaptive neighborhood to solve the curve fitting problem, where the approximating curve is represented as a piecewise Bézier curve. For experimental comparison, the discretization step size delta is set to twice the sample size. The local temperature loop is limited to 100 accepted solutions or 1000 iterations. The initial temperature is set to 100, and the minimum temperature is 1. The cooling factor is chosen as fixed at 0.9. The distance between two connecting points is no less than 4 and the minimum angle is 15°. Besides, the curve length and curve discrepancy contribute the same to the cost, and the initial number of knots is the same as AEFS.

5) **Grove** [31] employed a Bézier curve to regress the local data points and segmented the point set through the binary search (bisecting technique) beneath a prescribed fitting error. The bound for knot removal is relaxed to twice the error bound.
6) **Dung** [32] proposed fast knot calculation with two steps for non-uniform B-spline curves fitting. Their method also used the bisecting technique for coarse knot placement. The initial number of knots is varied around the setting in AEFS as well.

All algorithms are built and conducted utilizing quadratic splines or Bézier curves for a fair comparison while the parameters not mentioned above are initialized with the default system settings. The multiple-knot case is not discussed in this paper. Additionally, all experiments are carried out on the R platform, the Apple M1 Max GPU, 64GB RAM.

B. INFLECTION CURVES

We test the proposed method on a sequence of artificial data points of an inflection curve at the first beginning, which consists of two C^1 connected Bézier curve segments. In detail, two sequences of data points are generated from

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} 20 \\ 50 \end{pmatrix} (1-t)^2 + \begin{pmatrix} 40 \\ 100 \end{pmatrix} 2t(1-t) + \begin{pmatrix} 60 \\ 50 \end{pmatrix} t^2, \ t \in [0, 1]$$

and

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} 60 \\ 50 \end{pmatrix} (1-t)^2 + \begin{pmatrix} 80 \\ 0 \end{pmatrix} 2t(1-t) + \begin{pmatrix} 95 \\ 45 \end{pmatrix} t^2, \ t \in [0, 1]$$

separately. For the former segment, the parameter t is uniformly spaced by interval 0.025 producing 41 data points while the latter segment places interval 0.02 producing 51 data points. Omitting the overlapped point, the whole sequence consists of 91 data points.

Since the data points are derived from two C^1 connected Bézier curve segments, the ideal number of segments should be two after curve fitting. Meanwhile, the estimated control points should be close to the true control points, especially the connecting point between two fitted curves is supposed to get close to the real one. And apparently, the fitting error can not be small enough. The original points together with approximated curves obtained through all mentioned methods are shown in Fig.5 (a)-(h), respectively, and Table 1 displays some important fitting parameters and the experimental results where the latter includes the number of piece curves, the max and absolute average values of the discrepancy as well as the time cost.

As shown in the Fig.5 (h), the proposed method almost completely recovers the generated curve segments with the assistance of Bézier curve extension thought, especially the number of the segments and the position of the connecting point (the true position is indicated in black solid squares). We can easily see that the estimated curve number $m = 2$ is just right, the position of the approximated connecting point is pretty close to the true one, and no twist and overfitting happened in the fitted Bézier curve segments. In the contrast, the fitted curves using the methods from Smith, MaoZhao, Grove and Dung shows more curve segments than the ideal number in Fig. 5 (b)(c)(f)(g). And apart from the curve achieved by Grove, the residual three fitted curves are more sensitive than the one achieved in the proposed method, showing mild overfitting. For the fitted curves with two segments, the estimated connecting points are far from the true one as shown in Fig.5 (d)(e), and some unnecessary inflections are caused. Although the former fits better than the latter, both performance are worse than our proposed method whether it is the accuracy of connecting points or the fitting effect. Meanwhile, the fitting behaviors in Fig.5 (e)(f) are not as good as our proposed algorithm visually.

Combining with Table 1, we can see that the difference of discrepancy among models is in accordance with the illustration of all the fitted curves. The maximum discrepancy of our strategy is 0.1974, and the absolute average is 0.0364, both of which lay far less than the others. The increasing number of Bézier curve segments does not bring in higher fitting accuracy but causes overfitting for many models. Furthermore, AEFS predicts the position of the connecting point is $(60.1161, 49.8452)$, which is pretty close to the true $(60, 50)$, so are the other control points. Hence, the model performance of AEFS has been verified in various aspects in the case of inflection curves. In the following, the performance of all algorithms will be tested on some complex curves where the advantages of our method can be further verified.

C. COMPLEX CURVES

Right after the inflection curve, we evaluate the performance of AEFS on a complex butterfly curve, which can be depicted by a parametric deterministic function. This curve is generated using functions [38]:

$$\begin{cases} x(t) = 50 + 15\sin(t)\left(e^{\cos(t)} - 2\cos(4t) - \sin^5\left(\frac{t}{12}\right)\right) \\ y(t) = 40 + 15\cos(t)\left(e^{\cos(t)} - 2\cos(4t) - \sin^5\left(\frac{t}{12}\right)\right) \end{cases} \quad t \in [0, 2\pi].$$

In detail, the butterfly curve is sampled with 577 points at a uniform space $t = \frac{\pi}{288}$. The approximating curves from different methods together with the original data points are illustrated in Fig. 6. Table 2 lists parameter settings and experimental results. The number of interior knots strongly varies with the error threshold when the other parameters are kept unchanged. Here, to keep the fitting effect, we prescribe the initial error bound as 0.5 instead of 1 of the above case due to the more inflection of the butterfly curve.

Combining the fitted curve in Fig. 6 and the experimental results in Table 2, we foremost see that the fitted curves following MaoZhao's method and our method behave best. AEFS gives the maximum discrepancy (0.2091) which is slightly inferior to the performance of MaoZhao (0.1086).

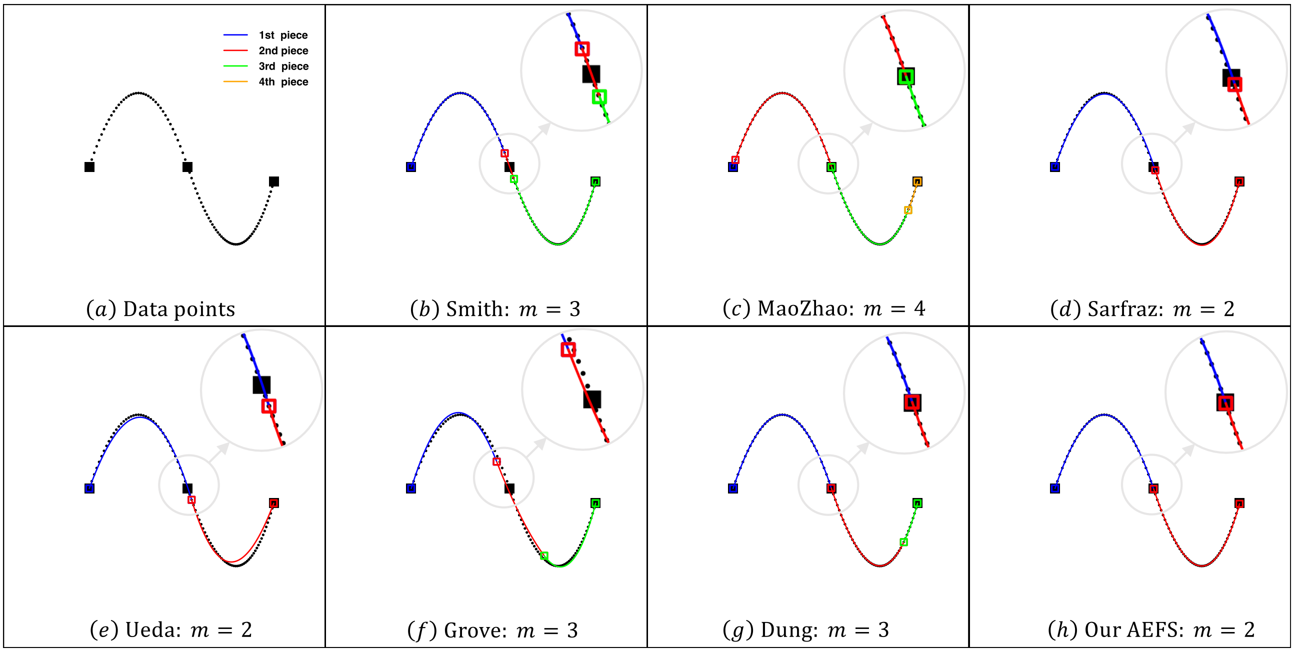

FIGURE 5. Fitted splines of inflection curves. The black solid circles represent the data points, and the black solid squares represent the true connecting points together with the endpoints of the sequence. The fitted curves using different methods are shown separately in lines, where the estimated curve number *m* follows the name at bottom, and each colored line represents one piece Bézier curve segment. And the hollow squares represent the estimated control points of each curve segment, where the closeness between them and the corresponding black solid squares indicates the fitting effect.

TABLE 1. The fitting results of the inflection curve using different methods.

Case	Fitting Parameters		Results			
	Initial segment No.	Error Bound	Execute Time	Segment No.	Max Fitting Error	Mean Error
Smith	20	1	0.172	3	0.3490	0.1090
MaoZhao	3	—	0.167	4	0.3285	0.0860
Sarfraz	—	1	0.076	2	0.7759	0.2859
Ueda	1	—	28.790	2	2.0637	0.7277
Grove	—	1	0.088	3	2.7810	0.8953
Dung	1	1	0.158	3	0.3650	0.0966
Our AEFS	—	1	0.388	**2**	**0.1974**	**0.0364**

TABLE 2. Butterfly curve results using different methods.

Case	Fitting Parameters		Results			
	Initial segment No.	Error Bound	Execute Time	Segment No.	Max Fitting Error	Mean Error
Smith	115	0.5	0.174	116	0.6052	0.0711
MaoZhao	40	—	87.668	43	**0.1086**	0.0514
Sarfraz	—	0.5	0.095	18	8.0153	3.6897
Ueda	37	—	37.092	38	3.1449	0.4959
Grove	—	0.5	0.105	36	1.4934	0.2305
Dung	40	0.5	4.478	34	1.5246	0.2812
Our AEFS	—	0.5	0.633	38	0.2091	**0.0361**

The absolute average discrepancies of both methods are still far less than the others, and our method exhibits the smallest error. Nonetheless, the time cost of MaoZhao's method is severely heavier than our method due to the optimization of the number of knots. Instead of the modified GCV, the problem of the heavy time cost may be alleviated by a better criterion.

The fitted curve achieved by the method of Smith is almost close to the original data points, except for several small areas with violent curvature changes. In the process of curve approximation, the knot removal is invalid as the maximum discrepancy is larger than the prescribed error bound of 0.5, which leads to less execution time and many more interior knots. It can be also highlighted that the number of interior

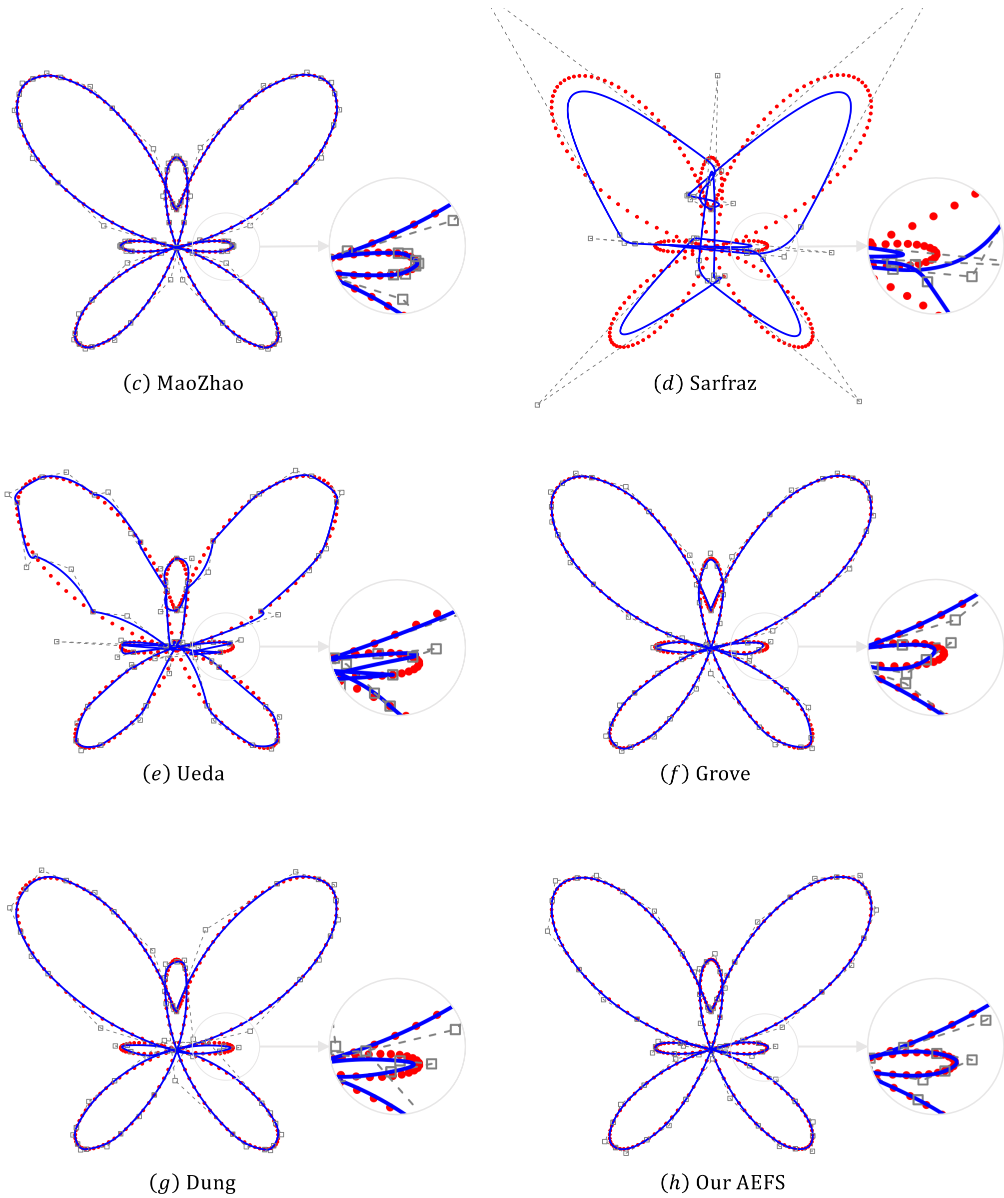

FIGURE 6. Fitted splines of butterfly curve. The red solid circles are the data points to be fitted, and the blue lines are the fitted curves by different methods. The grey hollow squares represent the control points, where the grey dotted lines connected them in sequential order.

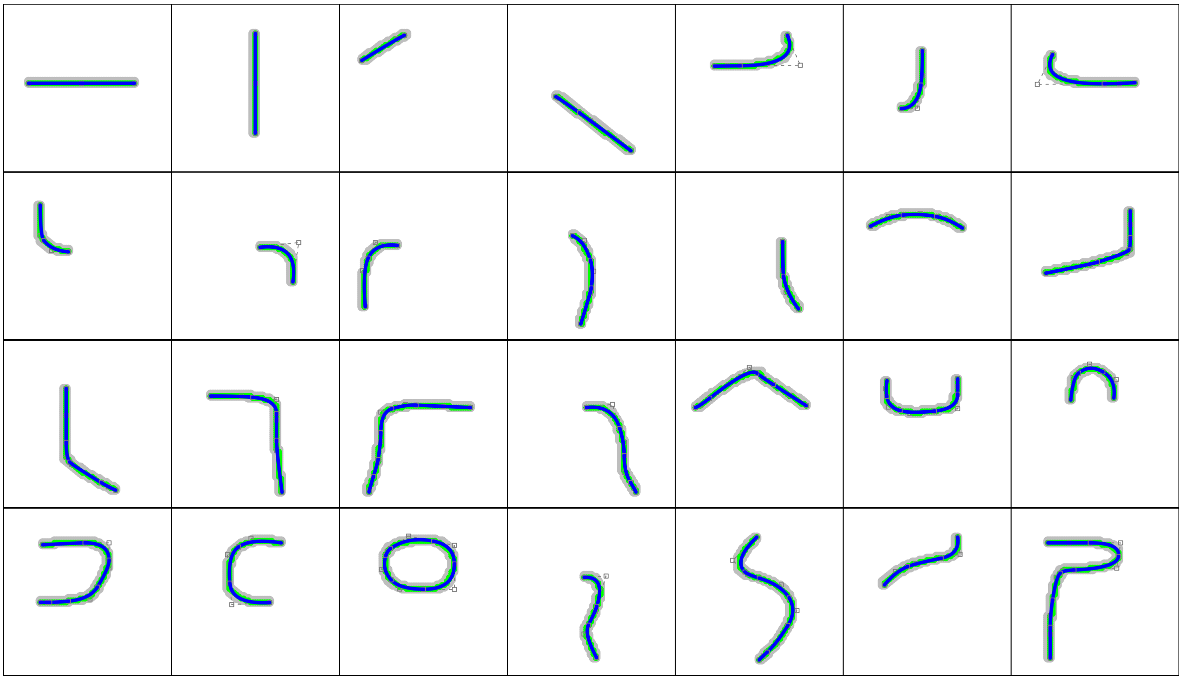

FIGURE 7. Fitted splines of typical seal strokes. The original seal strokes and the corresponding skeleton are delineated in grey and green, respectively. Above them, the solid blue curves depict the fitted spline with the control points indicated as grey squares.

knots of AEFS is appropriate. The fitted curves of Grove and Dung have the close number of interior knots as our strategy, but higher approximating errors frequently occur in the data points at dramatic inflection.

In this way, our proposed AEFS is notably faster due to the ingenious utilization of the geometrical properties and effectively approximating curves within a reasonable number of segments on all types of inflection as indicated. The obtained fitted curve is smooth enough to fully depict the original trends of the data points.

D. SEAL STROKES

The proposed AEFS has proved its fitting capability on both inflection curves and complex curves. In this section, we show that the proposed method is not only limited to generated curves but also any real-world shapes with unknown mathematical representation. The superb cost performance and concise curve trend guarantee its extensive applied scope.

Therefore, we further evaluated the model performance on the seal strokes that are the basic components of one type of ancient Chinese pictographs (seal characters). They are good examples to validate our method since seal strokes are represented by the coordinate sequence. Meanwhile, the proposed method can ingeniously vectorize raster images to vector images and finally reconstruct visually coherent vector representations.

Fig. 7 displays the main categories of seal strokes, including but not limited to line segments in different directions, arcs, curves, and loops. The original seal strokes represented by gray solid points are extracted from the character library of seal characters while the sequence of green points as the corresponding skeleton are the points to be fitted. We can realize the vectorization for seal strokes and the solid lines are produced by the obtained smoothly connected piecewise Bézier curves.

AEFS can flexibly depict the variation of the data sequence, where no matter whether gentle, violent changes or the transitions between these two can be processed appropriately. Overfitting rarely happens so that people can clearly recognize the fitted curve-represented seal strokes. In addition, the proposed method can be executed quickly and automatically once we set the universal error bound. Meanwhile, omitting the manual setting or the time-consuming searching for the number of knots also benefit the proposed method of batch implementation. Fig. 7 strongly demonstrates that AEFS is able to obtain excellent performance in real scenarios of curve approximation.

V. CONCLUSION

This work has successfully constructed an effective technology called adaptive extension fitting scheme (AEFS) for curve approximation utilizing piecewise Bézier curves that are widely used in many applications related to Computer Aided Geometric Design (CAGD) due to their simplicity and superb geometric properties. This proposed method is capable of adaptively locating connecting points between the Bézier curve segments. As the generalization of the Bézier approximation, AEFS not only inherits the geometrical properties of the Bézier curve but also adds local controllability to the fitted curve. To guarantee a reliable theoretical basis for AEFS, we first raise the criteria so as to strictly distinguish the Bézier curve and the Bézier curve segment from the mathematical perspective to avoid concept confusion. Subsequently, their intrinsic connection is explored in the manner of the extension theorem after introducing a standard representation of the Bézier curve segment. Inspired by the theorem, we construct the adaptive strategy for piecewise Bézier curve fitting with the assistance of curve extension, resulting in appropriate connecting points together with well-fitted Bézier curves. The quantity and location of the connecting points can be detected and determined adaptively, which saves massive time consumption in tedious parameter optimization. The usage of the extension theorem enhances the fitting process effectively. Hence our proposed method has sufficient advantages in both algorithm design and mathematical theory.

Moreover, the experimental results demonstrate that our strategy achieves superior fitting performance to other reported curve approximation methods. It is evident that our algorithm yields a lesser number of segments to represent most of the shapes than any other methods mentioned. The authenticity of shape contours for the empirical shapes is quite significantly better than other algorithms concerning high fitting accuracy in the meanwhile. Both theories and experiments present that AEFS can approximate the curves with outstanding performance. This adaptive strategy can further be applied in extensive scopes, such as capturing the concise curve trend of character strokes.

AEFS has successfully shown its effectiveness in curve approximation. However, it requires more fitted curve segments once the actual trend of data points fluctuates violently, which inevitably complicates the curve expression. A feasible treatment is to utilize the Bézier curves with a higher degree for an appropriate fit, yet the constraints on the connecting points should be cautiously reconsidered to ensure the continuity between adjacent segments, or it may result in unexpected sudden changes in the curve trend. Meanwhile, the constraints are supposed to be imposed in an adaptive way following the trend of the data points to better approximate the point sequence. We will consider and discuss the practical application of strategies and techniques that can be improved in further study.

REFERENCES

[1] H. Prautzsch, W. Boehm, and M. Paluszny, *Bézier and B-Spline Techniques*, vol. 6. Berlin, Germany: Springer, 2002.

[2] E. Sobrinho, R. Sanomya, R. Ueda, H. Tiba, M. D. S. G. Tsuzuki, J. C. Adamowski, E. C. N. Silva, R. C. Carbonari, and F. Buiochi, "Development of a methodology for evaluation of a structural damage in turbine blades from hydropower generators," in *Proc. 20th COBEM*. Gramado, Brazil: Citeseer, 2009, pp. 1–10.

[3] S. A. Juliano, "Nonlinear curve fitting: Predation and functional response curves," in *Design and Analysis of Ecological Experiments*. Boca Raton, FL, USA: Chapman & Hall/CRC, 2020, pp. 159–182.

[4] N. Sampathraja, L. A. Kumar, R. S. Kumar, and I. M. Wartana, "Solar power forecasting using adaptive curve-fitting algorithm," in *Proc. Int. Conf. Artif. Intell., Smart Grid Smart City Appl. (AISGSC)*. Cham, Switzerland: Springer, 2020, pp. 227–236.

[5] L. Piegl and W. Tiller, "Algorithm for approximate nurbs skinning," *Comput.-Aided Design*, vol. 28, no. 9, pp. 699–706, Sep. 1996.

[6] T. Maekawa, Y. Matsumoto, and K. Namiki, "Interpolation by geometric algorithm," *Comput.-Aided Design*, vol. 39, no. 4, pp. 313–323, Apr. 2007.

[7] S.-I. Gofuku, S. Tamura, and T. Maekawa, "Point-tangent/point-normal B-spline curve interpolation by geometric algorithms," *Comput.-Aided Design*, vol. 41, no. 6, pp. 412–422, Jun. 2009.

[8] S. Okaniwa, A. Nasri, H. Lin, A. Abbas, Y. Kineri, and T. Maekawa, "Uniform B-spline curve interpolation with prescribed tangent and curvature vectors," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 9, pp. 1474–1487, Sep. 2012.

[9] P. Bézier, *The Mathematical Basis of the UNIURF CAD System*. Oxford, U.K.: Butterworth-Heinemann, 2014.

[10] S. Baydas and B. Karakas, "Defining a curve as a Bezier curve," *J. Taibah Univ. Sci.*, vol. 13, no. 1, pp. 522–528, Dec. 2019.

[11] Y. Mineur, T. Lichah, J. M. Castelain, and H. Giaume, "A shape controled fitting method for Bézier curves," *Comput. Aided Geometric Design*, vol. 15, no. 9, pp. 879–891, Oct. 1998.

[12] P. Pandunata and S. M. H. Shamsuddin, "Differential evolution optimization for Bézier curve fitting," in *Proc. 7th Int. Conf. Comput. Graph., Imag. Visualizat.*, Aug. 2010, pp. 68–72.

[13] A. Y. Hasegawa, R. S. Rosso Jr., and M. S. G. Tsuzuki, "Bézier curve fitting with a parallel differential evolution algorithm," *IFAC Proc. Volumes*, vol. 46, no. 7, pp. 233–238, 2013.

[14] M. Misro, A. Ramli, and J. Ali, "Quintic trigonometric Bézier curve with two shape parameters," *Sains Malaysiana*, vol. 46, no. 5, pp. 825–831, May 2017.

[15] S. Bibi, M. Abbas, M. Y. Misro, and G. Hu, "A novel approach of hybrid trigonometric Bézier curve to the modeling of symmetric revolutionary curves and symmetric rotation surfaces," *IEEE Access*, vol. 7, pp. 165779–165792, 2019.

[16] S. Bibi, M. Abbas, K. T. Miura, and M. Y. Misro, "Geometric modeling of novel generalized hybrid trigonometric Bézier-like curve with shape parameters and its applications," *Mathematics*, vol. 8, no. 6, p. 967, Jun. 2020.

[17] M. Ammad, M. Y. Misro, and A. Ramli, "A novel generalized trigonometric Bézier curve: Properties, continuity conditions and applications to the curve modeling," *Math. Comput. Simul.*, vol. 194, pp. 744–763, Apr. 2022.

[18] S. Zain, M. Misro, and K. T. Miura, "Curve fitting using generalized fractional Bézier curve," *Comput.-Aided Design Appl.*, vol. 20, no. 2, pp. 350–363, 2023.

[19] M. Banks and E. Cohen, "Real time spline curves from interactively sketched data," *ACM SIGGRAPH Comput. Graph.*, vol. 24, no. 2, pp. 99–107, Mar. 1990.

[20] T. Pavlidis, "Curve fitting with conic splines," *ACM Trans. Graph.*, vol. 2, no. 1, pp. 1–31, Jan. 1983.

[21] L. Piegl, "Interactive data interpolation by rational Bezier curves," *IEEE Comput. Graph. Appl.*, vol. CGA-7, no. 4, pp. 45–58, Apr. 1987.

[22] M. Plass and M. Stone, "Curve-fitting with piecewise parametric cubics," in *Proc. 10th Annu. Conf. Comput. Graph. Interact. Techn.*, 1983, pp. 229–239.

[23] Z. Qiao, M. Hu, Z. Tan, Z. Liu, L. Liu, and W. Hu, "An accurate and fast method for computing offsets of high degree rational Bézier/NURBS curves with user-definable tolerance," *J. Comput. Lang.*, vol. 52, pp. 1–9, Jun. 2019.

[24] H.-T. Yau and J.-B. Wang, "Fast Bezier interpolator with real-time lookahead function for high-accuracy machining," *Int. J. Mach. Tools Manuf.*, vol. 47, no. 10, pp. 1518–1529, Aug. 2007.

[25] E. K. Ueda, A. K. Sato, T. C. Martins, R. Y. Takimoto, R. S. U. Rosso, and M. S. G. Tsuzuki, "Curve approximation by adaptive neighborhood simulated annealing and piecewise Bézier curves," *Soft Comput.*, vol. 24, no. 24, pp. 18821–18839, Dec. 2020.

[26] D. I. S. Adi, S. M. B. Shamsuddin, and S. Z. M. Hashim, "NURBS curve approximation using particle swarm optimization," in *Proc. 7th Int. Conf. Comput. Graph., Imag. Visualizat.*, Aug. 2010, pp. 73–79.

[27] L. Piegl and W. Tiller, *The NURBS Book*. Berlin, Germany: Springer, 1996.

[28] P. De Casteljau, *Mathématiques et Cao, volume 2: Formes à Pôles*. Paris, Londres, Lausanne, 1985.

[29] S. D. Pande, U. A. Patil, R. Chinchore, and M. S. Rani Chetty, "Precise approach for modified 2 stage algorithm to find control points of cubic Bezier curve," in *Proc. 5th Int. Conf. Comput., Commun., Control Autom. (ICCUBEA)*, Sep. 2019, pp. 1–8.

[30] P. J. Schneider, "An algorithm for automatically fitting digitized curves," *Graph. gems*, vol. 1, pp. 612–626, Aug. 1990.

[31] O. Grove, K. Rajab, L. A. Piegl, and S. Lai-Yuen, "From CT to NURBS: Contour fitting with B-spline curves," *Comput.-Aided Design Appl.*, vol. 8, no. 1, pp. 3–21, Jan. 2011.

[32] V. T. Dung and T. Tjahjowidodo, "A direct method to solve optimal knots of B-spline curves: An application for non-uniform b-spline curves fitting," *PLoS ONE*, vol. 12, no. 3, Mar. 2017, Art. no. e0173857.

[33] P. L. Smith, "Curve fitting and modeling with splines using statistical variable selection techniques," Langley Res. Center, Hampton, VA, usa, NASA Rep. 166034, 1982.

[34] W. Mao and L. H. Zhao, "Free-knot polynomial splines with confidence intervals," *J. Roy. Stat. Soc. B, Stat. Methodology*, vol. 65, no. 4, pp. 901–919, Nov. 2003.

[35] C. De Boor, *Spline Toolbox: For Use With MATLAB: User's Guide*. Natick, MA, USA: MathWorks, 1992.

[36] M. Sarfraz, "Approximation with B-splines curves," in *Interactive Curve Modeling: With Applications to Computer Graphics, Vision and Image Processing*. London, U.K.: Springer, 2008, pp. 173–194.

[37] R. R. Patterson, "Projective transformations of the parameter of a bernstein-Bézier curve," *ACM Trans. Graph.*, vol. 4, no. 4, pp. 276–290, Oct. 1985.

[38] T. H. Fay, "The butterfly curve," *Amer. Math. Monthly*, vol. 96, no. 5, pp. 442–443, May 1989.

XINGYU CUI received the bachelor's degree in statistics from Beijing Normal University, Beijing, China, in 2016, where she is currently pursuing the Ph.D. degree in applied statistics with the School of Statistics.

Her current research interests include pattern recognition, character recognition, curve fitting, and statistical learning.

YONG LI received the M.S. and Ph.D. degrees in probability theory and mathematical statistics from Beijing Normal University, Beijing, China, in 1988 and 1991, respectively.

He is currently a Professor and the Director of the Academic Committee, School of Statistics, Beijing Normal University. He has authored/coauthored 11 books and over 90 research papers in scientific journals and conference proceedings. His research interests include diffusion processes, statistical learning, generalized linear models, and character recognition. He is the Deputy Editor-in-Chief of the *High School Mathematics* [Textbook (A) Edition, (People's Education Press)].

LILI XU received the M.S. degree in statistics from Texas Tech University, Lubbock, TX, USA, in 2010, and the Ph.D. degree in computer science from the University of Macau, Macau, China, in 2021.

She is currently a Lecturer with the School of Applied Mathematics, Beijing Normal University, Zhuhai, China. Her research interests include statistical machine learning, Bayesian optimization, and related applications in computational intelligence.