Computer-Aided Design 34 (2002) 415–419
COMPUTER-AIDED DESIGN
www.elsevier.com/locate/cad

Technical note

An extension algorithm for B-splines by curve unclamping ☆

Shi-Min Hu[a,*], Chiew-Lan Tai[b], Song-Hai Zhang[a]

[a]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[b]Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong

Received 18 September 2000; revised 8 February 2001; accepted 22 February 2001

Abstract

This paper presents an algorithm for extending B-spline curves and surfaces. Based on the unclamping algorithm for B-spline curves, we propose a new algorithm for extending B-spline curves that extrapolates using the recurrence property of the de Boor algorithm. This algorithm provides a nice extension, with maximum continuity, to the original curve segment. Moreover, it can be applied to the extension of B-spline surfaces. Extension to both single and multiple target points/curves are considered in this paper. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: B-spline curve; Extension; de Boor algorithm

1. Introduction

B-spline is among the most commonly used method for curve and surface design, and it has been widely used in practical CAD systems [1,2]. In a system that uses B-spline or NURBS as a shape design tool, we should implement many practical algorithms, such as position and derivatives evaluation, knot insertion, knot deletion and degree elevation. Based on these fundamental algorithms, we can perform geometric modelling and processing, such as interpolation, fitting and fairing of curves and surfaces, constructing sweep or lofting surfaces, computing intersection of curves and surfaces, shape modification and set operation of solids with free-form surfaces.

Besides the above-mentioned functions, which a CAD system usually provides, the extension of a given B-spline curve or surface is also a useful function. A natural problem is to extend a B-spline curve $P(t)$ to a given point R, and to represent the extended curve in B-spline form. In many systems, a common solution is to add a Bézier curve with GC^1 continuity at the end of the B-spline curve, and then convert the entire curve into B-spline form. Shetty and White [3] proposed a practical and straightforward method for the extension of rational B-spline curves and surfaces without modifying the shape and the parametrization of the original curves and surfaces. With their method, the extended curve is represented as a B-spline curve with knot vector in the form of $\{0,0,0,0,\ldots 1,1,1,1,s,s,s,s\}$, where the order of the curve is assumed to be four and s is a number greater than 1. However, a B-spline curve with knot vector in this form is not favoured in CAD applications.

The objective of this paper is to develop an algorithm for extrapolating B-spline curves and surfaces using curve unclamping and the recurrence property of de Boor algorithm. The paper is organised as follows. The definition of the B-spline curves, the de Boor algorithm, and the unclamping algorithm are described in Section 2. Section 3 presents the extrapolation method for the extension of B-spline curves. Section 4 investigates extension of B-spline surfaces. Conclusion is given in Section 5.

2. B-spline curves, de Boor algorithm, and curve unclamping algorithm

A B-spline curve of order k with control points $P_i (i = 0,1,\ldots,n)$ can be defined as:

$$P(t) = \sum_{i=0}^{n} P_i N_{i,k}(t),\ t_{k-1} \le t \le t_{n+1} \tag{1}$$

where $N_{i,k}(t)$ are the B-spline basis functions of order k defined over the knot vector $T = \{t_0,t_1,\ldots,\ t_k,\ldots,\ t_n,\ldots,\ t_{n+k}\}$, and can be defined by the well-known de Boor-Cox formula [1,2].

☆ Part of this research was conducted while the first author was a visiting researcher at the Hong Kong University of Science & Technology.

* Corresponding author. Tel.: +86-10-627-82052; fax: +86-10-627-82025.

E-mail address: shimin@tsinghua.edu.cn (S.-M. Hu).

0010-4485/02/$ - see front matter © 2002 Elsevier Science Ltd. All rights reserved.
PII: S0010-4485(01)00108-7

$$N_{i,1}(t) = \begin{cases} 1, & t_i \le t < t_{i+1} \\ 0, & \text{Otherwise} \end{cases}$$

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t). \quad (2)$$

A point on the curve $P(t)$ at parameter $t(t \in [t_j, t_{j+1}])$ can be computed using the following de Boor algorithm.

$$P_i^r(t) = \begin{cases} P_i, \ r = 0, i = 0, 1, \ldots, n \\ \dfrac{t_{i+k} - t}{t_{i+k} - t_{i+r}} P_i^{r-1} + \dfrac{t - t_{i+r}}{t_{i+k} - t_{i+r}} P_{i+1}^{r-1}, \\ r = 1, 2, \ldots, k-1; i = j-k+1, \ldots, j-r \end{cases} \quad (3)$$

and $P(t) = P_{j-k+1}^{k-1}(t)$. Fig. 1 shows the computational process for a fourth order curve.

Knot vectors of B-splines can be classified as clamped and unclamped [2]. For clamped knot vectors, it is common to use the following form for order k B-spline curves:

$$T_1 : \underbrace{0, 0, \ldots, 0}_{k}, t_k, \ldots, t_n, \underbrace{1, 1, \ldots, 1}_{k}. \quad (4)$$

The first part of our B-spline extension algorithm involves unclamping B-spline curves from T_1 to:

$$T_2 : \underbrace{0, 0, \ldots, 0}_{k}, t_k, \ldots, t_n, t_{n+1}, \ldots, t_{n+k},$$

where:

$$t_{n+k} \ge t_{n+k-1} \ge \ldots \ge t_{n+1} \ge 1 \quad (5)$$

The following unclamping algorithm proposed by Piegl and Tiller (see page 577 of Ref. [2]) can be applied to compute the new control points $Q_i (i = 0,1,\ldots,n)$ with unclamped knot vector T_2.

Algorithm 1: Curve unclamping on the right side[1]

1. Set initial value. Let:

$$Q_i^0 = Q_i, \ i = n-k+2, \ n-k+3, \ldots, n.$$

2. Let:

$$\begin{cases} Q_i^r = Q_i^{r-1}, \ i = n-k+2, \ n-k+3, \ldots, n-r \\ Q_i^r = \dfrac{Q_i^{r-1} - (1-\alpha) Q_{i-1}^r}{\alpha}, \ i = n-r+1, n-r+2, \ldots, n \\ \text{where } \alpha = \dfrac{t_{n+1} - t_i}{t_{i+r+1} - t_i} \end{cases}$$

$r = 1,2,\ldots,k-2$.

3. Let:

$$Q_i = \begin{cases} P_i, \ i = 0, 1, \ldots, \ n-k+1 \\ Q_i^{k-2}, \ i = n-k+2, \ n-k+3, \ldots, n \end{cases}$$

3. Extension of B-spline curves

3.1. Extending a B-spline curve to a target point

For a given B-spline curve $P(t)$, we now show that it is easy to extend the curve to a target point R by using the unclamping algorithm. The target point R may be some user-defined point or be automatically generated by methods such as those described in Shetty and White [3]. For a given parameter $t_r > 1$, their method computes the target point by reflecting point $P(1 - t_r)$ about the normal of the curve at $P(1)$.

We first estimate the parametrization of the extended curve. After adding one target point, the knot vector of $P(t)$ would be changed from T_1 to:

$$T_3 : \underbrace{0, 0, \ldots, 0}_{k}, t_k, \ldots, t_n, 1, \underbrace{u, u, \ldots, u}_{k}, \quad (6)$$

where $u > 1$. We calculate u according to the chord length estimation; that is, u is determined by:

$$u = 1 + \frac{\|P_n - R\|}{\sum_{r=0}^{n-k+1} \|P(t_{k+r}) - P(t_{k+r-1})\|} \quad (7)$$

where $\|\cdot\|$ is the Euclidean norm.

Once the parametrization is estimated, we change the knot vector of $P(t)$ from T_1 to

$$\underbrace{0, 0, \ldots, 0}_{k}, t_k, \ldots, t_n, 1, \underbrace{u, u, \ldots, u}_{k-1}$$

and use Algorithm 1 to compute the new control points Q_i, $0 \le i \le n$.

The new curve after the extension can be represented as:

$$\tilde{P}(t) = \sum_{i=0}^{n+1} P_i N_{i,k}(t), \ 0 \le t \le u \quad (8)$$

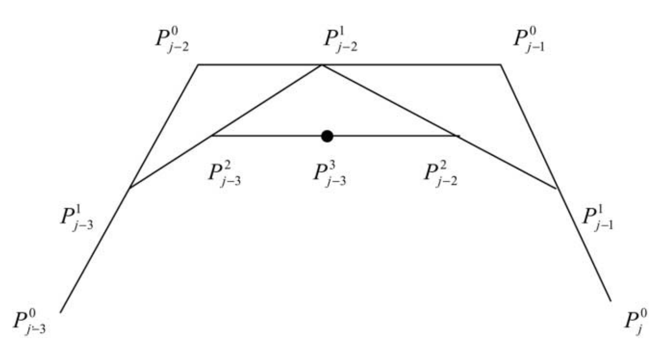

Fig. 1. Computational process of de Boor algorithm for $k = 4$.

[1] The original algorithm in Ref. [2] unclamps curves on both the left and right sides, and k denotes degree instead of order.

where:

$$P_i = \begin{cases} Q_i, & i = 0, 1, \ldots, n; \\ R, & i = n+1 \end{cases}.$$

By the properties of the B-spline basis functions, we can rewrite the knot vector T_3 as:

$$\underbrace{0, 0, \ldots, 0}_{k}, \frac{t_k}{u}, \ldots, \frac{t_n}{u}, \frac{1}{u}, \underbrace{1, 1, \ldots, 1}_{k}.$$

Fig. 2 shows the effects of extending an order 4 B-spline curve: (a) shows the original curve and the target point, (b) is the same curve after curve unclamping and (c) is the new curve after extension.

3.2. Extension with multiple target points

We will only consider the extension of B-spline curves with two target points; the discussion for the case of more than two target points is similar.

Let the two target points be R and $\hat{R}$. After the extension, the knot vector of the new curve would be changed from T_1 to:

$$T_4 : \underbrace{0, 0, \ldots, 0}_{k}, t_k, \ldots, t_n, 1, u, \underbrace{v, v, \ldots, v}_{k}, \quad (9)$$

where $v > u > 1$. We calculate u and v according to the chord length estimation, i.e., u is defined by Eq. (7), and v is determined by:

$$v = u + \frac{\left\|\hat{R} - R\right\|}{\sum_{r=0}^{n-k+1} \|P(t_{k+r}) - P(t_{k+r-1})\|} \quad (10)$$

By unclamping the knot vector of $P(t)$ from T_1 to:

$$\underbrace{0, 0, \ldots, 0}_{k}, t_k, \ldots, t_n, 1, u, \underbrace{v, v, \ldots, v}_{k-2},$$

we use Algorithm 1 to compute the new control points Q_i, $0 \le i \le n$, of the reparametrized curve $Q(t)$. We first consider how to extend $Q(t)$ to the first target point R to obtain a curve with knot vector:

$$\underbrace{0, 0, \ldots, 0}_{k}, t_k, \ldots, t_n, 1, u, \underbrace{v, v, \ldots, v}_{k-1}.$$

The extension from R to $\hat{R}$ is then similar to the procedure described in Section 3.1.

Note that $P(t)$ is a B-spline curve of order k, thus we can achieve C^{k-2} continuity between the previous curve segment (at endpoint $Q(1) = P_n$) and the extended segment, which has endpoints $Q(1)$ and R. In fact, the control points of the previous segment are Q_i, $i = n-k+1, n-k+2, \ldots, n$, and the control point of the new segment would be Q_i, $i = n-k+2, n-k+3, \ldots, n+1$; that is, only one new control point Q_{n+1} has to be computed.

We propose a new extrapolation algorithm based on the inverse process of de Boor algorithm. To illustrate the idea of this algorithm, we take order four as an example. In Fig. 1, let $j = n+1$ and P^3_{j-3} be the target point R. We want to compute the unknown control point P^0_j from the given interpolated point R and the known control points P^0_{j-3}, P^0_{j-2} and P^0_{j-1}. The following two computation steps can achieve this goal:

1. Compute the intermediate points P^1_{j-3}, P^1_{j-2},P^2_{j-3} by de Boor recurrence, i.e., compute P^1_{j-3} from P^0_{j-3} and P^0_{j-2}, P^1_{j-2} from P^0_{j-2} and P^0_{j-1}, and P^2_{j-3} from P^1_{j-3} and P^1_{j-2}.
2. Compute the points P^2_{j-2}, P^1_{j-1},P^0_j by the inverse process of de Boor recurrence, i.e., compute P^2_{j-2} from P^2_{j-3} and $P^3_{j-3} = R$, P^1_{j-1} from P^1_{j-2} and P^2_{j-2}, then $Q_{n+1} = P^0_j$ from P^0_{j-1} and P^1_{j-1}

The recursive algorithm for the general case of extrapolating order k B-spline curves is as follows.

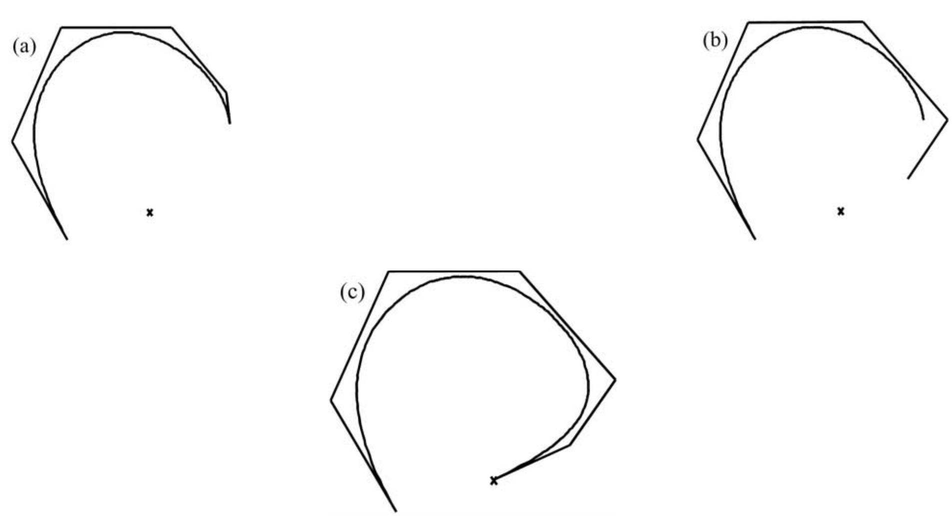

Fig. 2. Extension of an order four B-spline curve to a target point.

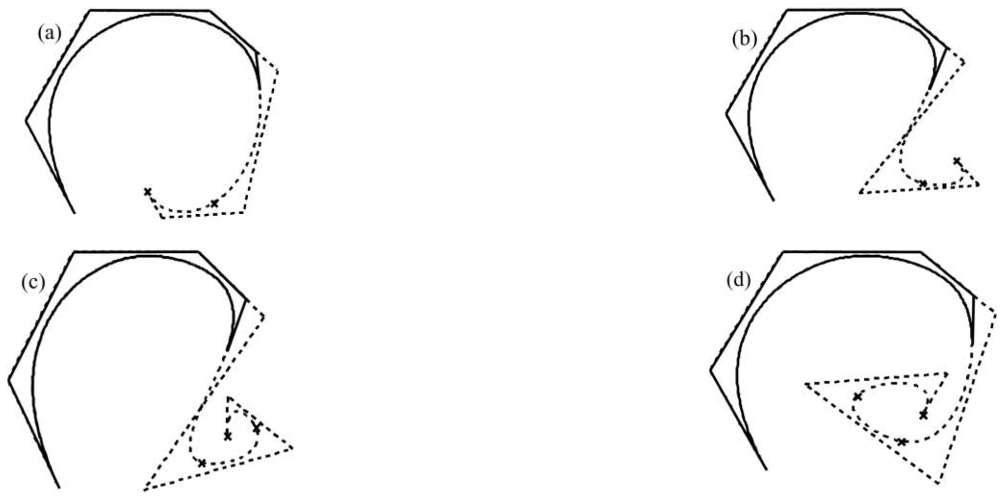

Fig. 3. Extension of B-spline curves of order four with multiple target points.

Algorithm 2: Computation of Q_{n+1} by extrapolation

1. Set initial value. Let $j = n + 1, t = t_{j+1}, Q_{j-k+1}^{k-1}(t) = R$ and $Q^{0}_{l+(j-k+1)}(t) = Q_{n+2-k+l}$ for $l = 0,1,\ldots,k-2$.
2. De Boor recurrence. Let:

$$Q_i^r(t) = \frac{t_{i+k} - t}{t_{i+k} - t_{i+r}} Q_i^{r-1}(t) + \frac{t - t_{i+r}}{t_{i+k} - t_{i+r}} Q_{i+1}^{r-1}(t),$$

$$r = 1, 2, \ldots, k-2;\; i = j-k+1,\; \ldots, j-r-1$$

3. Inverse de Boor recurrence:

$$Q_i^r(t) = \frac{(t_{i+k} - t_{i+r})Q_{i+1}^{r+1}(t) - (t_{i+k} - t)Q_{i+1}^{r}}{t - t_{i+r}},$$

$$r = k-2, k-3, \ldots, 0;\; i = j - r$$

4. $Q_{n+1} = Q_j^0(t)$.

After computing Q_{n+1} by Algorithm 2, we only have to let $Q_{n+2} = \hat{R}$, and the new curve can be represented as:

$$Q(t) = \sum_{i=0}^{n+2} Q_i N_{i,k}(t),\; 0 \le t \le 1. \tag{11}$$

with the knot vector:

$$\underbrace{0, 0, \ldots, 0}_{k}, \frac{t_k}{v}, \ldots, \frac{t_n}{v}, \frac{1}{v}, \frac{u}{v}, \underbrace{1, 1, \ldots, 1}_{k}$$

Fig. 3 shows the results of extending several B-spline curves to two and three target points. The extended curve segments are shown as dotted lines while the original curve segments are shown as solid lines.

4. Extension of B-spline surfaces

An order $k \times l$ B-spline surface $P(t,s)$ with control points $P_{ij}(0 \leqslant i \leqslant n, 0 \leqslant j \leqslant m)$ can be defined as:

$$P(t,s) = \sum_{i=0}^{n}\sum_{j=0}^{m} P_{ij} N_{ik}(t) N_{jl}(s),\; t_{k-1} \le t \le t_{n+1}, s_{l-1} \le s \le s_{m+1}, \tag{12}$$

where $N_{i,k}(t)$ and $N_{j,l}(s)$ are the B-spline basis functions of order k and l, respectively, defined over knot vectors T_1 and:

$$T_5 : \underbrace{0, 0, \ldots, 0}_{l}, s_l, \ldots, s_m, \underbrace{1, 1, \ldots, 1}_{l}.$$

We now consider the extension of $P(t,s)$ in the direction of t to two target curves $T_1(s)$ and $T_2(s)$. Without loss of generality[2], we assume that $T_1(s)$ and $T_2(s)$ are order l curves with knot vector T_5 and control points P_j^1 and $P_j^2 (j = 0, 1, \ldots, m)$ respectively.

Due to the tensor-product structure of B-spline surfaces, it suffices to extend the $m + 1$ B-spline curves:

$$P_j(t) = \sum_{i=0}^{n} P_{ij} N_{ik}(t),\; j = 0, 1, \ldots, m$$

to target points P_j^1 and P_j^2 with a common parametrization. Since the knot vector T_1 should be changed to T_4 after the extension, we compute u and v by averaging $u_j(j = 0,1,\ldots,m)$ and $v_j(j = 0,1,\ldots,m)$, respectively, where u_j and v_j are computed by Eqs. (7) and (10) respectively.

Fig. 4 shows the result of extending a B-spline surface with two target curves. The extended surface patches are shown in dotted lines while the original surface patch and target curves are shown in solid lines.

[2] If not, we can use degree elevation and knot insertion algorithms to achieve it.

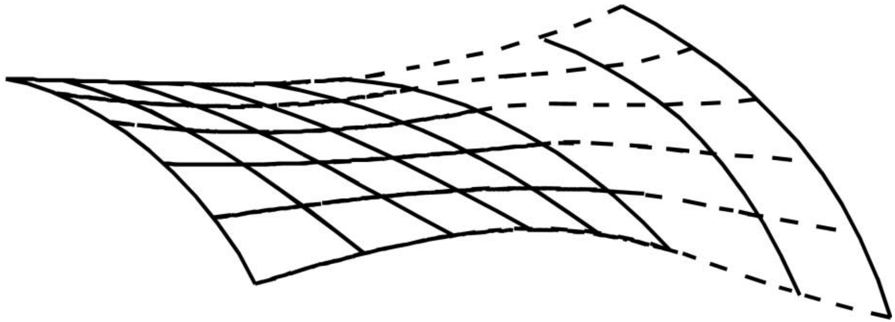

Fig. 4. Extension of a B-spline surface of order 4×4 with two target curves.

5. Conclusion

This paper presents a new method for extending B-spline curves and surfaces. We first extend B-spline curves to one or more target points by curve unclamping and extrapolation using the recurrence property of de Boor algorithm. The algorithm for curves is then shown to be applicable for extending B-spline surfaces to single or multiple target curves. Experiment results have shown that the proposed method is efficient and more practical than the method by Shetty and White [3].

Acknowledgements

This research was partially supported by the Natural Science Foundation of China (Project Number 69902004), NKBRSF (Project Number 1998030600) and the Hong Kong Research Grant Council (HKUST6215/99E). We especially thank the reviewers for their helpful comments.

References

[1] Farin G. Curves and surfaces for computer aided geometric design. 4th ed. San Diego: Academic Press, 1997.
[2] Piegl L, Tiller W. The NURBS book. 2nd ed. Berlin: Springer-Verlag, 1997.
[3] Shetty S, White PR. Curvature-continuous extensions for rational B-spline curves and surfaces. Computer-Aided Design 1991;23-7:484–91.

Shi-Min Hu is an associate professor in Department of Computer Science and Technology at Tsinghua University. He received the BS in mathematics from Jilin university of China in 1990, the MS and the PhD in CAGD and Computer Graphics from Zhejiang university of China in 1993 and 1996, he finished his postdoctor research at Tsinghua University in 1998. His current research intersets are in computer-aided geometric design, computer graphics. A list of recent publications can be found at http://ncc.cs.tsinghua.edu.cn/~shimin.

Chiew-Lan Tai received her BSc and MSc in Mathematics from The University of Malaya, and her MSc in Computer and Information Sciences from National University of Singapore (NUS). She pursued her PhD from The University of Tokyo on a fellowship awarded by Japan Society for the Promotion of Science (JSPS). She joined The Hong Kong University of Science & Technology, Department of Computer Science, as an Assistant Professor in August 1997. Her research interests include geometric modeling, computer graphics, and scientific visualization.

Song-Hai Zhang is a graduate student in the Department of Computer Science and Technology, Tsinghua University, Beijing, P. R. China, where he received a BS in 2001. His research interests are computer aided geometric design and computer graphics.