ELSEVIER

Computer-Aided Design 35 (2003) 893–899

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Approximate merging of B-spline curves via knot adjustment and constrained optimization

Chiew-Lan Tai[a], Shi-Min Hu[b,*], Qi-Xing Huang[b]

[a]*Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong, People's Republic of China*
[b]*Department of Computer Science and Technology, Tsinghua University, Beijing 100084, People's Republic of China*

Received 5 March 2002; received in revised form 18 September 2002; accepted 30 September 2002

Abstract

This paper addresses the problem of approximate merging of two adjacent B-spline curves into one B-spline curve. The basic idea of the approach is to find the conditions for precise merging of two B-spline curves, and perturb the control points of the curves by constrained optimization subject to satisfying these conditions. To obtain a merged curve without superfluous knots, we present a new knot adjustment algorithm for adjusting the end k knots of a kth order B-spline curve without changing its shape. The more general problem of merging curves to pass through some target points is also discussed.

Keywords: B-spline curves; Merging; Knot adjustment; Constrained optimization

1. Introduction

Approximate conversion is an important issue in data communication between different CAD systems [1]. As mentioned by Hoschek [2], approximate conversion includes the following two problems:

- *Degree reduction*: finding a parametric curve of degree n that approximates the given curve of degree m ($n < m$).
- *Merging*: merging as many curve segments of degree n as possible to get one curve segment of degree m ($n \le m$).

Degree reduction methods for Bézier, Ball, and B-spline curves and surfaces have been extensively investigated [3–13]. Merging is one of the main methods for data reduction; by merging as many curve segments as possible into one curve, the amount of geometric data needed for communication can be reduced. In Ref. [14], we presented a method for approximate merging of a pair of Bézier curves using constrained optimization method. The objective of this paper is to consider approximate merging of B-spline curves.

The approximate merging problem we address is as follows. Given two adjacent order k B-spline curves $\mathbf{P}(t)$ and $\mathbf{R}(s)$ with knot vectors $\mathbf{T} = \{t_0, t_1, \ldots, t_k, \ldots, t_n, \ldots, t_{n+k}\}$ and $\mathbf{S} = \{s_0, s_1, \ldots, s_k, \ldots, s_m, \ldots, s_{m+k}\}$ respectively, and control points $\mathbf{P}_i$ $(i = 0, 1, \ldots, n)$ and $\mathbf{R}_j$ $(j = 0, 1, \ldots, m)$ respectively, find an order k B-spline curve $\mathbf{F}(u)$with control points $\mathbf{F}_i$ $(i = 0, 1, \ldots, n + m - k + 2)$ and knot vector $\mathbf{U} = \{t_0, t_1, \ldots, t_k, \ldots, t_n, t_{n+1} = s'_{k-1}, s'_k, \ldots, s'_{m+k}\}$, where $s'_i = f(s_i)$ for some linear function f, such that a suitable distance function $d(\mathbf{F}, \bar{\mathbf{F}})$ between $\mathbf{F}(u)$ and

$$\bar{\mathbf{F}}(u) = \begin{cases} \mathbf{P}(u), & t_{k-1} \le u \le t_{n+1} \\ \mathbf{R}(f^{-1}(u)), & s'_{k-1} \le u \le s'_{m+1} \end{cases},$$

is minimized.

The basic idea of our method is to first find the conditions for precise merging of two B-spline curves, and perturb the control points of the curves by constrained optimization subject to satisfying the precise merging conditions. We then reparametrize the resulting curves using a new *knot adjustment* algorithm we present. Such knot adjustment is needed for producing a merged B-spline curve efficiently without superfluous knots.

The remainder of the paper is organized as follows. Section 2 presents the definition of the B-spline curves, the de Boor algorithm, and our knot adjustment algorithm. Section 3 describes the method for approximate merging of a pair of B-spline curves; the more general problem of

* Corresponding author. Tel.: +86-10-627-82052; fax: +86-10-627-82025.
E-mail address: shimin@tsinghua.edu.cn (S.-M. Hu).

0010-4485/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
PII: S0010-4485(02)00176-8

approximate merging with point constraints is also discussed. Conclusions are given in Section 4.

2. B-spline curves, evaluation and knot adjustment

A B-spline curve of order k with control points $\mathbf{P}_i$ $(i = 0, 1, \ldots, n)$ can be defined as

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

where $N_{i,k}(t)$are the B-spline basis functions of order k defined over the knot vector $\mathbf{T} = \{t_0, t_1, \ldots, t_k, \ldots t_n, \ldots, t_{n+k}\}$, which is defined by the following recursive de Boor-Cox formula [15]

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

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

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

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

and $\mathbf{P}(t) = \mathbf{P}_{j-k+1}^{k-1}(t)$.

To compute the derivatives of a B-spline curve, by letting $P_i^0 = P_i$, and writing

$$\mathbf{P}^{(0)}(t) = \mathbf{P}(t) = \sum_{i=0}^{n} \mathbf{P}_i^0 N_{i,k}(t),$$

we get (see p. 97 in Ref. [15])

$$\mathbf{P}^{(l)}(t) = \sum_{i=0}^{n-l} \mathbf{P}_i^l N_{i+1,k-l}(t), \tag{4}$$

where

$$\mathbf{P}_i^l = \begin{cases} \mathbf{P}_i, & l = 0 \\ \dfrac{k-l}{t_{i+k} - t_{i+l}} (\mathbf{P}_{i+1}^{l-1} - \mathbf{P}_i^{l-1}), & l > 0 \end{cases}. \tag{5}$$

Suppose two kth order B-spline curves $\mathbf{P}(t)$ and $\mathbf{R}(s)$with knot vectors $\mathbf{T} = \{t_0, t_1, \ldots, t_k, \ldots, t_n, \ldots, t_{n+k}\}$ and $\mathbf{S} = \{s_0, s_1, \ldots, s_k, \ldots, s_m, \ldots, s_{m+k}\}$ are to be merged. In our merging algorithm, after perturbing the control points to satisfy precise merging conditions, we adjust the knot vectors of the curves. Specifically, we adjust the last k knots of the curve $\mathbf{P}(t)$ so that they match k knots of the curve $\mathbf{R}(s)$; that is, the knot vector of $\mathbf{P}(t)$ must be adjusted from $\mathbf{T}$ to $\mathbf{T}' = \{t_0, t_1, \ldots, t_k, \ldots, t_n, t_{n+1}, s_k, s_{k+1}, \ldots, s_{2k-2}\}$. Although this knot adjustment can be achieved by first clamping $\mathbf{T}$, and then unclamping it to $\mathbf{T}'$ [15], for efficiency, we propose a new general algorithm to directly adjust the knot vector $\mathbf{T}$ to $\mathbf{T}_2 = \{t_0, t_1, \ldots, t_k, \ldots, t_n, t_{n+1}, t'_{n+2}, \ldots, t'_{n+k}\}$, where $t_{n+1} \le t'_{n+2} \le \cdots \le t'_{n+k-1} \le t'_{n+k}$.

Suppose the new curve defined on the new knot vector $\mathbf{T}_2$ is denoted by $\mathbf{Q}(t)$with control points $\mathbf{Q}_0, \mathbf{Q}_1, \ldots, \mathbf{Q}_n$. Since the shape of the curve remains unchanged, we have $\mathbf{P}(t) = \mathbf{Q}(t)$. We claim that $\mathbf{Q}_i = \mathbf{P}_i$, $0 \le i \le n-k+2$, and the other $\mathbf{Q}_i, n-k+3 \le i \le n$ can be computed recursively by Algorithm 1. The correctness proof is given in Appendix 1.

Algorithm 1. Computing the new control points after knot adjustment from $\mathbf{T}$ to $\mathbf{T}_2$

1. Compute $\mathbf{P}_{n-k+2}^l$, $l = 1, 2, \ldots, k-1$ by Eq. (5).
2. Let $\mathbf{Q}_{n-k+2}^l = \mathbf{P}_{n-k+2}^l$, for $l = 0, 1, \ldots, k-1$.
3. Compute $\mathbf{Q}_i^0$ for $n-k+3 \le i \le n$ by (see top row of Fig. 1)

$$\mathbf{Q}_i^l = \frac{t'_{i+k-1} - t_{i+l}}{k-l-1} \mathbf{Q}_{i-1}^{l+1} + \mathbf{Q}_{i-1}^l, \tag{6}$$

$$i = n-k+3, \quad n-k+4, \ldots, n, \quad l = 0, 1, \ldots, n-i,$$

which can be derived from Eq. (5)

4. Let $\mathbf{Q}_i = \mathbf{P}_i$ for $i \le n-k+2$, and $\mathbf{Q}_i = \mathbf{Q}_i^0$ for $n-k+3 \le i \le n$.

Fig. 2 shows the results of adjusting the knot vectors of two cubic B-spline curves. The original control polygons are shown in solid line, while the new control polygons are shown in dotted line. In Fig. 2(a), the knot vector is adjusted from $\{0, 0, 0, 0, 0.5, 1, 1, 1, 1\}$ to $\{0, 0, 0, 0, 0.5, 1, 1.3, 1.3, 1.3\}$. In Fig. 2(b), is adjusted to $\{-0.6, -0.4, -0.2, 0, 0.5, 1, 1.5, 1.6, 1.7\}$.

3. Merging by constrained optimization

Suppose $\mathbf{P}(t)$ and $\mathbf{R}(s)$ are two B-spline curves to be approximately merged. Let $\mathbf{R}(s)$ be a B-spline curve with control points $\mathbf{R}_i$ $(i = 0, 1, \ldots, m)$ and knot vector

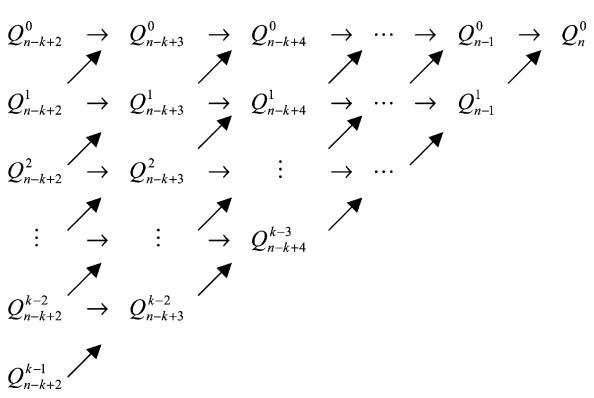

Fig. 1. Recursive computation of $\mathbf{Q}_i = \mathbf{Q}_i^0$, $i = n-k+2, \ldots, n$.

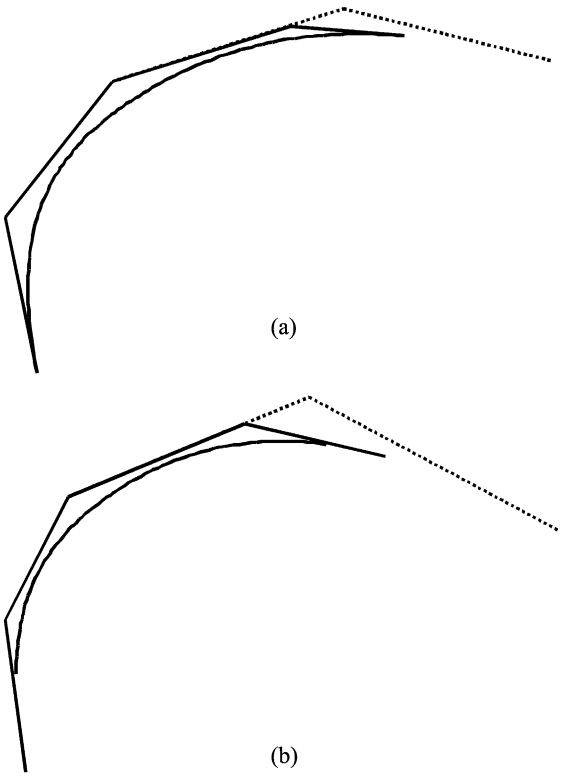

Fig. 2. Two examples of knot adjustment of B-spline curves.

$\mathbf{S} = \{s_0, s_1, \ldots, s_k, \ldots, s_m, \ldots, s_{m+k}\}$. Without loss of generality, we suppose that the order of $\mathbf{R}(s)$ is also k, and $t_{n+1} = s_{k-1}$. To obtain a good parametrization for the merged curve, we perform a linear transformation f on the knot vector $\mathbf{S}$ such that

$$\frac{\text{arc length of } \mathbf{P}(t) \text{ over } [t_n, t_{n+1}]}{\text{arc length of } \mathbf{R}(s) \text{ over } [s_k - s_{k-1}]} = \frac{t_{n+1} - t_n}{s_k - s_{k-1}}.$$

3.1. Conditions for precise merging

We first derive the conditions for the curves $\mathbf{P}(t)$ and $\mathbf{R}(s)$ to be precisely merged into one B-spline curve. We let the curves share common derivatives, that is, to precisely merge at $\mathbf{P}(t_{n+1}) = \mathbf{R}(s_{k-1})$, we need

$$\mathbf{P}^{(l)}(t_{n+1}) = R^{(l)}(s_{k-1}), \qquad l = 0, 1, \ldots, k-2,$$

i.e.

$$\sum_{i=n-k+1}^{n-l} \mathbf{P}_i^l N_{i+1,k-l}^T(t_{n+1}) = \sum_{i=0}^{k-1-l} \mathbf{R}_i^l N_{i+1,k-l}^S(s_{k-1}), \tag{7}$$

$$l = 0, 1, \ldots, k-2,$$

where $N_{i,k}^T(t)$ $N_{i,k}^S(s)$ are B-spline basis functions defined on the knot vectors $\mathbf{T}$ and $\mathbf{S}$, respectively. According to Eq. (6), $\mathbf{P}_i^l$ and $\mathbf{R}_i^l$ in Eq. (7) can be rewritten as $\sum_{j=n-k+1}^{n} a_{ij}^{(l)} \mathbf{P}_j$ and $\sum_{j=0}^{k-1} b_{ij}^{(l)} \mathbf{R}_j$, respectively, where $a_{ij}^{(l)}$ and $b_{ij}^{(l)}$ can be computed by Algorithms 2 and 3, respectively.

Algorithm 2. Computing $a_{ij}^{(l)}$, $n-k+1 \le i \le n-l$, $n-k+1 \le j \le n$, $0 \le l \le k-2$.

1. For $n-k+1 \le j \le n$, let $p_0, p_1, \ldots, p_n$ be scalars, and

$$p_i = \delta_{ij} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}.$$

2. Let

$$p_i^l = \begin{cases} p_i, & \text{for } l = 0 \\ \dfrac{k-l}{t_{i+k} - t_{i+l}} (p_{i+1}^{l-1} - p_i^{l-1}) & \text{for } l > 0 \end{cases}.$$

3. Let $a_{ij}^{(l)} = p_i^l$, $n-k+1 \le i \le n-l$, $0 \le l \le k-2$.

Algorithm 3. Computing $b_{ij}^{(l)}$, $0 \le i \le k-1-l$, $0 \le j \le k-1$, $0 \le l \le k-2$.

1. For $0 \le j \le k-1$, let $p_0, p_1, \ldots, p_n$ be scalars and

$$p_i = \delta_{ij} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}.$$

2. Let

$$p_i^l = \begin{cases} p_i, & \text{for } l = 0 \\ \dfrac{k-l}{s_{i+k} - s_{i+l}} (p_{i+1}^{l-1} - p_i^{l-1}), & \text{for } l > 0 \end{cases}.$$

3. Let $b_{ij}^{(l)} = p_i^l$, $0 \le i \le k-1-l$, $0 \le l \le k-2$.

Then the precise merging conditions in Eq. (7) can be rewritten as

$$\sum_{j=n-k+1}^{n} \left(\sum_{i=n-k+1}^{n-l} a_{ij}^{(l)} N_{i+1,k-l}^T(t_{n+1}) \right) \mathbf{P}_j - \sum_{j=0}^{k-1} \left(\sum_{i=0}^{k-1-l} b_{ij}^{(l)} N_{i+1,k-l}^S(S_{k-1}) \right) \mathbf{R}_j = 0, \tag{8}$$

$$l = 0, 1, \ldots, k-2.$$

3.2. Merging by constrained optimization

To merge two arbitrary curves $\mathbf{P}(t)$ and $\mathbf{R}(s)$, we first perturb their control points so that the curves can be precisely merged. We achieve this by minimizing the total perturbation of the control points subject to the precise merging constraints Eq. (8). Let the perturbation of the control points of $\mathbf{P}(t)$ and $\mathbf{R}(s)$ be denoted by $\boldsymbol{\epsilon}_i = \{\epsilon_i^x, \epsilon_i^y, \epsilon_i^z\}$ $i = n-k+2, n-k+3, \ldots, n$ and $\delta_i = \{\delta_i^x, \delta_i^y, \delta_i^z\}$,

$i=0,1,\ldots,k-2$ respectively.[1] That is, the modified curves are

$$\hat{\mathbf{P}}(t)=\sum_{i=0}^{n}\hat{\mathbf{P}}_i N_{i,k}^{T}(t)=\sum_{i=0}^{n}(\mathbf{P}_i+\boldsymbol{\epsilon}_i)N_{i,k}^{T}(t),\quad t_{k-1}\le t\le t_{n+1},$$

and

$$\hat{\mathbf{R}}(s)=\sum_{i=0}^{m}\hat{\mathbf{R}}_i N_{i,k}^{S}(s)=\sum_{i=0}^{m}(\mathbf{R}_i+\delta_i)N_{i,k}^{S}(s),\quad s_{k-1}\le s\le s_{m+1},$$

and the precise-merging conditions from Eq. (8) become

$$\sum_{j=n-k+1}^{n}\left(\sum_{i=n-k+1}^{n-l}a_{ij}^{(l)}N_{i+1,k-l}^{T}(t_{n+1})\right)(\mathbf{P}_j+\boldsymbol{\epsilon}_j)-\sum_{j=0}^{k-1}\left(\sum_{i=0}^{k-1-l}b_{ij}^{(l)}N_{i+1,k-l}^{S}(s_{k-1})\right)(\mathbf{R}_j+\delta_j)=0,\tag{9}$$

$l=0,1,\ldots,k-2.$

More generally, we consider the approximate merging problem in which the merged curve is constrained to pass through some target points on the original curves. This can be achieved by adding point constraint conditions as follows

$$\sum_{i=n-k+1}^{n}\mathbf{P}_i N_{i,k}^{T}(t_j)=\sum_{i=n-k+1}^{n}(\mathbf{P}_i+\boldsymbol{\epsilon}_i)N_{i,k}^{T}(t_j),\quad j=0,1,\ldots,g.$$

$$\sum_{i=0}^{k-1}\mathbf{R}_i N_{i,k}^{S}(s_j)=\sum_{i=0}^{k-1}(\mathbf{R}_i+\delta_i)N_{i,k}^{S}(s_j),\quad j=0,1,\ldots,h.$$

where t_j, $j=0,1,\ldots,g$ and s_j, $j=0,1,\ldots,h$ are parameters of the target points on $\mathbf{P}(t)$ and $\mathbf{R}(s)$, respectively.[2] So we have

$$\sum_{i=n-k+1}^{n}\boldsymbol{\epsilon}_i N_{i,k}^{T}(t_j)=0,\quad j=0,1,\ldots,g.\tag{10}$$

$$\sum_{i=0}^{k-1}\delta_i N_{i,k}^{S}(s_j)=0,\quad j=0,1,\ldots,h.\tag{11}$$

We determine $\boldsymbol{\epsilon}_i$ δ_i by setting the optimization objective $\mathrm{O}(\boldsymbol{\epsilon}_i,\delta_j)$ as

$$\mathrm{Min}\,\mathrm{O}(\boldsymbol{\epsilon}_i,\delta_j)=\sum_{i=n-k+2}^{n}\boldsymbol{\epsilon}_i+\sum_{j=0}^{k-2}\delta_j,\tag{12}$$

and define the Lagrange function as

$$L=\sum_{i=n-k+2}^{n}\boldsymbol{\epsilon}_i+\sum_{j=0}^{k-2}\delta_j+\sum_{l=0}^{k-2}\lambda_l\left(\sum_{j=n-k+1}^{n}\left(\sum_{i=n-k+1}^{n-l}a_{ij}^{(l)}N_{i+1,k-l}^{T}(t_{n+1})\right)\times(\mathbf{P}_j+\boldsymbol{\epsilon}_j)-\sum_{j=0}^{k-1}\left(\sum_{i=0}^{k-1-l}b_{ij}^{(l)}N_{i+1,k-l}^{S}(s_{k-1})\right)(\mathbf{R}_j+\delta_j)\right)+\sum_{j=0}^{g}\lambda_{k-1+j}\left(\sum_{i=n-k+1}^{n}\boldsymbol{\epsilon}_i N_{i,k}^{T}(t_j)\right)+\sum_{j=0}^{h}\lambda_{k+g+j}\left(\sum_{i=0}^{k-1}\delta_i N_{i,k}^{S}(s_j)\right),\tag{13}$$

where $\lambda_i=[\lambda_i^x,\lambda_i^y,\lambda_i^z]$ are the Lagrange multipliers. By setting $\partial L/\partial\boldsymbol{\epsilon}_i^x$, $\partial L/\partial\boldsymbol{\epsilon}_i^y$, $\partial L/\partial\boldsymbol{\epsilon}_i^z$, $\partial L/\partial\delta_j^x$, $\partial L/\partial\delta_j^y$, $\partial L/\partial\delta_j^z$, $\partial L/\partial\lambda_i^x$, $\partial L/\partial\lambda_i^y$, $\partial L/\partial\lambda_i^z$ to zero, and writing the derived equations in vector form, we obtain a system of linear equations, which includes Eqs. (9)–(11), and the following two equations

$$2\boldsymbol{\epsilon}_i=-\sum_{l=0}^{k-2}\sum_{j=n-k+1}^{n-l}\lambda_l a_{ij}^{(l)}N_{i+1,k-l}^{T}(t_{n+1})+\sum_{j=0}^{g}\lambda_{k-1+j}N_{i,k}^{T}(t_j),\tag{14}$$

$n-k+2\le i\le n,$

$$2\delta_j=\sum_{l=0}^{k-2}\sum_{i=0}^{k-1-l}\lambda_l b_{ij}^{(l)}N_{i+1,k-l}^{S}(s_{k-1})+\sum_{j=0}^{h}\lambda_{k+g+j}N_{i,k}^{S}(s_j),\tag{15}$$

$0\le j\le k-2.$

By solving the linear system, the constrained optimization solution can be obtained.

After obtaining two modified curves that can be precisely merged, we let

$$\mathbf{T}_{\mathrm{new}}=\{t_0,t_1,\ldots,t_k,\ldots,t_n,t_{n+1}=s_{k-1},s_k,s_{k+1},\ldots,s_m,\ldots,s_{m+k}\}$$

be the knot vector of the merged curve to be constructed. Using Algorithm 1, we adjust the knot vector $\mathbf{T}$ of curve $\hat{\mathbf{P}}(t)$ to $\mathbf{T}'=\{t_0,t_1,\ldots,t_k,\ldots,t_n,t_{n+1},s_k,s_{k+1},\ldots,s_{2k-2}\}$ and the knot vector $\mathbf{S}$ of curve $\hat{\mathbf{R}}(s)$ to $\mathbf{S}'=\{t_{n-k+2},\ldots,t_n,t_{n+1}=s_{k-1},s_k,s_{k+1},\ldots,s_{m+k}\}$, respectively. For simplicity, we still denote the control points after knot adjustment as $\hat{\mathbf{P}}_i$, $i=0,1,\ldots,n$; $\hat{\mathbf{R}}_i$, $i=0,1,\ldots,m$. Since $\hat{\mathbf{P}}(t)$ and $\hat{\mathbf{R}}(s)$ satisfy the precise merging conditions, i.e. they have common derivatives at parameter $t_{n+1}=s_{k-1}$, we conclude that the control points of the merged curve are

$$\{\hat{\mathbf{P}}_0,\hat{\mathbf{P}}_1,\ldots,\hat{\mathbf{P}}_{n-k+1},\hat{\mathbf{P}}_{n-k+2}=\hat{\mathbf{R}}_0,\hat{\mathbf{P}}_{n-k+3}=\hat{\mathbf{R}}_1,\ldots,\hat{\mathbf{P}}_n=\hat{\mathbf{R}}_{k-2},\hat{\mathbf{R}}_{k-1},\ldots,\hat{\mathbf{R}}_m\}.$$

3.3. Error estimation and examples

B-spline curves of order 4 are among the most commonly used parametric curves in shape design. Here, we give an error analysis for merging two B-spline curves of order 4; error analysis for order 2 and 3 curves can be similarly derived. For higher-order curves, it seems hard to obtain explicit estimation.

[1] From precise merging conditions, only $k-1$ control points are involved in each curve, see Eq. (7). Thus, for convenience in subsequent derivation, we just set $\boldsymbol{\epsilon}_i=0$ $(i=0,1,\ldots,n-k+1)$ and $\delta_i=0$ $(i=k-1,k-2,\ldots,m)$.

[2] Note that the total number of point constraints for two curves should be less than $k-1$, otherwise the solution of the equation system does not exist.

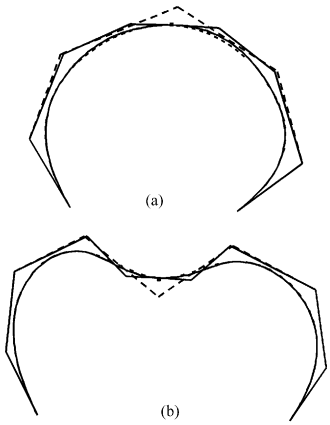

Fig. 3. Approximate merging of two clamped B-spline curves.

Theorem 1. *Suppose* $\mathbf{P}(t)$ *and* $\mathbf{R}(s)$ *are two fourth-order B-spline curves, and* $t_{n+1} = \cdots = t_{n+k}$, $s_0 = \cdots = s_{k-1}$, *we have the following estimation*

$$\|\mathbf{P}-\mathbf{F}\| \le \frac{3}{4}\frac{u_1\Delta_1}{u_1+\Delta_1}\|\mathbf{d}_1\| + \frac{1}{3}\frac{(u_1+\Delta_1)^2+u_1\Delta_1+u_1^2}{(u_1+\Delta_1)^2} \times(u_1\Delta_1\|\mathbf{d}_2\| + |u_1-\Delta_1|\|\mathbf{d}_1\|), \tag{16}$$

$$\|\mathbf{R}-\mathbf{F}\| \le \frac{3}{4}\frac{u_1\Delta_1}{u_1+\Delta_1}\|\mathbf{d}_1\| + \frac{1}{3}\frac{(u_1+\Delta_1)^2+u_1\Delta_1+u_2^2}{(u_1+\Delta_1)^2} \times(u_1\Delta_1\|\mathbf{d}_2\| + |u_1-\Delta_1|\|\mathbf{d}_1\|), \tag{17}$$

where $u_1 = t_{n+1} - t_n$, $\Delta_1 = s_k - s_{k-1}$, $\mathbf{d}_1 = \mathbf{P}^{(1)}(t_{n+1}) - \mathbf{R}^{(1)}(t_{k-1})$, $\mathbf{d}_2 = \mathbf{P}^{(2)}(t_{n+1}) - \mathbf{R}^{(2)}(s_{k-1})$, $\|\mathbf{P}-\mathbf{F}\| = \max_{t\in[t_{k-1},t_{n+1}]}\|\mathbf{P}(t)-\mathbf{F}(t)\|$, *and* $\|\mathbf{R}-\mathbf{F}\| = \max_{s\in[s_{k-1},s_{n+1}]}\|\mathbf{R}(s)-\mathbf{F}(t_{n+1}+s)\|$.

A proof of Theorem 1 is given in Appendix B.

We give some examples to illustrate our method. Figs. 3 and 4 show some examples of merging clamped and unclamped B-spline curves, respectively. The original curves are rendered in solid line, and the new merged curves are rendered in dotted line. In Fig. 3(a), the knot vectors of the original curves are both [0,0,0,0,0.5,1,1,1,1], and that of the merged curve is [0,0,0,0,0.5,1,1.48588,1.972, 1.972,1.972,1.972], and in Fig. 3(b), the knot vectors of the original curves are both [0,0,0,0,0.33,0.67,1,1,1,1], and

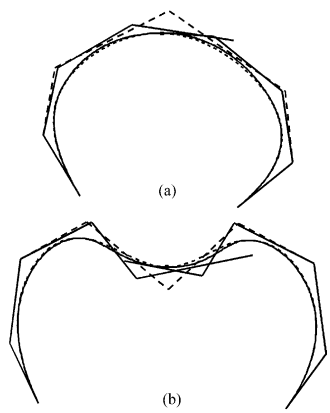

Fig. 4. Approximate merging of two unclamped B-spline curves.

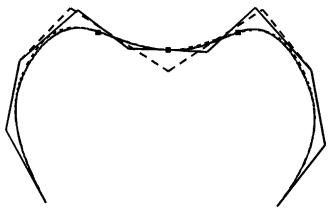

Fig. 5. Approximate merging with point constraints.

that of the merged curve is [0,0,0,0,0.33,0.67, 1,1.338,1.675, 2.012,2.012,2.012,2.012]. In Fig. 4(a), the original knot vectors are [0,0,0,0,0.5,1,1.2,1.4,1.6] and [0,0, 0,0,0.5,1,1,1,1], and the resulting one is [0,0,0,0,0.5,1,1.523, 2.045,2.045,2.045,2.045]. In Fig. 4(b), the original knot vectors are [0,0,0,0,0.33,0.67,1,1.2,1.4,1.6] and [−0.3, − 0.2, − 0.1,0,0.33,0.67,1,1,1,1], and the resulting one for the merged curve is [0,0,0,0,0.33,0.67,1,1.342, 1.684,2.025, 2.025,2.025,2.025].

To increase the accuracy of merging, we can add geometric constraints, i.e. constraining the resulting curve to pass through some target points on the original curves. Fig. 5 shows the effect of merging with point constraints, the knot vectors of the original two curves are both [0,0,0,0,0.33,0.67,1,1,1,1], the constrained points are $P(0.75)$ and $Q(0.25)$, which are marked as squares in figure.

4. Conclusion and future works

This paper presents an algorithm for approximate merging of two B-spline curves. We derive the precise merging conditions by letting the curves share common derivatives. The curves are modified by constrained optimization to satisfy the precise merging conditions. By applying the knot adjustment algorithm, the control points of the merged curve can be directly obtained from those of the curves satisfying precise merging conditions. The resulting merged curve has no superfluous knots.

We have used ‘Discrete coefficient norm’ in L_2 sense in this paper, as in Ref. [14], but merging with the ‘squared difference integral norm’ is also possible. As a future work, merging of multiple B-spline curves and of surfaces can be further investigated.

Acknowledgements

This research was partially supported by the Natural Science Foundation of China (Project Number 60225016), NKBRSF (Project Number 1998030600) and the Hong Kong Research Grant Council (HKUST6215/99E).

Appendix A. Correctness proof of the knot adjustment algorithm

By the recursive definition of $\mathbf{P}_i^l$ and $\mathbf{Q}_i^l$ in Eqs. (5) and (6), respectively, it is sufficient to prove that (I) $\mathbf{Q}_i = \mathbf{P}_i$ for

$i \le n-k+1$; (II) $\mathbf{Q}^l_{n-k+2} = \mathbf{P}^l_{n-k+2}$ for $l = 0, 1, \ldots, k-2$. We prove them for two possible cases.

(a) $n \ge 2k-2$, i.e. at least k curve segments.

From the local property of B-spline curves, for $t \in [t_s, t_{s+1})$, only k basis functions $N_{i,k}(t)$, $i = s-k+1, \ldots, s$ are non-zero, i.e.

$$\mathbf{P}(t) = \sum_{i=s-k+1}^{s} \mathbf{P}_i N_{i,k}(t), \qquad t \in [t_s, t_{s+1}). \tag{A1}$$

Let $\hat{N}_{i,k}(t)$ denote the B-spline basis functions defined on the knot vector T_2. From the construction of the knot vectors T and T_2, we have $N_{i,k}(t) = \hat{N}_{i,k}(t)$, $s-k+1 \le i \le s$, for all $s \le n-k$.

Consider the first B-spline curve segment in $[t_{k-1}, t_k]$. We have $N_{i,k}(t) = \hat{N}_{i,k}(t)$, $0 \le i \le k-1$ (note $n \ge 2k-2$); hence we get $\mathbf{Q}_i = \mathbf{P}_i$, $0 \le i \le k-1$. By applying the same argument to the intervals $t \in [t_s, t_{s+1}]$ for $k-2 \le s \le n-k+1$. we have $\mathbf{Q}_i = \mathbf{P}_i$ for $i \le n-k+1$. Thus (I) is true.

Next we show that (II) is true. We first consider the first derivative of the B-spline curves. For the segment $[t_{n-k+2}, t_{n-k+3}]$, $\mathbf{P}(t) = \mathbf{Q}(t)$ yields $\mathbf{P}^{(1)}(t) = \mathbf{Q}^{(1)}(t)$. Hence, from Eq. (4), we have

$$\sum_{i=n-2k+3}^{n-k+1} \mathbf{P}^1_i N_{i,k-1}(t) = \sum_{i=n-2k+3}^{n-k+1} \mathbf{Q}^1_i \hat{N}_{i,k-1}(t).$$

By applying the same argument as for segment $t \in [t_{k-1}, t_k]$ above, it is obvious that $N_{i,k-1}(t) = \hat{N}_{i,k-1}(t)$ for $n-2k+3 \le i \le n-k+1$. We therefore conclude that $\mathbf{Q}^1_{n-k+1} = \mathbf{P}^1_{n-k+1}$. Analogously, for subsequent segments, $[t_{n-k+3}, t_{n-k+4}], \ldots, [t_n, t_{n+1}]$, we have $\mathbf{Q}^l_{n-k+1} = \mathbf{P}^l_{n-k+1}$, $l = 0, 1, \ldots, k-1$. Since

$$\mathbf{Q}^l_{n-k+2} = \frac{t'_{n+1} - t_{n-k+2+l}}{k-l-1} \mathbf{Q}^{l+1}_{n-k+1} + \mathbf{Q}^l_{n-k+1},$$

$l = 0, 1, \ldots, k-2$,

$$\mathbf{P}^l_{n-k+2} = \frac{t_{n+1} - t_{n-k+2+l}}{k-l-1} \mathbf{P}^{l+1}_{n-k+1} + \mathbf{P}^l_{n-k+1},$$

$l = 0, 1, \ldots, k-2$,

and $t'_{n+1} = t_{n+1}$, we conclude $\mathbf{Q}^l_{n-k+2} = \mathbf{P}^l_{n-k+2}$, $l = 0, 1, \ldots, k-2$.

(b) $n < 2k-2$, i.e. fewer than k segments.

From the equations

$$\mathbf{Q}^l_{i+1} = \frac{t'_{i+k} - t_{i+l+1}}{k-l-1} \mathbf{Q}^{l+1}_i + \mathbf{Q}^l_i, \qquad l = 0, 1, \ldots, k-2,$$

$$\mathbf{P}^l_{i+1} = \frac{t_{i+k} - t_{i+l+1}}{k-l-1} \mathbf{P}^{l+1}_i + \mathbf{P}^l_i, \qquad l = 0, 1, \ldots, k-2,$$

and $(t'_{i+k} - t_{i+l+1})/(k-l-1) = (t_{i+k} - t_{i+l+1})/(k-l-1)$ for $i \le n-k+1$, we know that, to prove (I) $\mathbf{Q}_i = \mathbf{P}_i$ for $i \le n-k+1$ and (II) $\mathbf{Q}^l_{n-k+2} = \mathbf{P}^l_{n-k+2}$ for $l = 0, 1, \ldots, k-2$,, it is sufficient to show that $\mathbf{Q}^l_0 = \mathbf{P}^l_0$ ($0 \le l \le k-1$).

We first prove $\mathbf{Q}_0 = \mathbf{P}_0$. For the B-spline curves $\mathbf{P}(t)$ and $\mathbf{Q}(t)$, we insert multi-knots in the spans $[t_n, t_{n+1}]$ and $[t'_n, t'_{n+1}]$, respectively, for $2k-2-n$ times, and obtain a new curve $\bar{P}(t)$ with control points $\bar{P}_i$ ($0 \le i \le 2k-2$), and another new curve $\bar{\mathbf{Q}}(t)$ with control points $\bar{Q}_i$ ($0 \le i \le 2k-2$). Since knot insertion does not change the shape of the curves, we have $\bar{\mathbf{P}}(t) = \bar{\mathbf{Q}}(t)$. Investigating $\bar{\mathbf{P}}(t)$ and $\bar{\mathbf{Q}}(t)$ from the viewpoint of knot adjustment, we conclude that $\bar{\mathbf{P}}_0 = \bar{\mathbf{Q}}_0$ by part (a) of the correctness proof. From the knot insertion formula by Boehm (see p. 141 in Ref. [15]), we have $\mathbf{P}_0 = \bar{\mathbf{P}}_0$, $Q_0 = \bar{\mathbf{Q}}_0$, thus $\mathbf{P}_0 = \mathbf{Q}_0$. By similar discussion for the curves $\mathbf{P}^{(l)}(t)$ and $\mathbf{Q}^{(l)}(t)$, $l = 1, 2, \ldots, k-1$, we have $\mathbf{Q}^l_0 = \mathbf{P}^l_0$ ($1 \le l \le k-1$). This completes the proof.

Appendix B. Proof of Theorem 1

Proof. Let the perturbation of the control points $\mathbf{P}_{n-2}$, $\mathbf{P}_{n-1}, \mathbf{P}_n$ in curve $\mathbf{P}(t)$ and $\mathbf{R}_0, \mathbf{R}_1, \mathbf{R}_2$ in $R(s)$ be $\epsilon_1, \epsilon_2, \epsilon$ and $\epsilon\,\delta_1, \delta_2$, respectively. Due to the precise merging condition, we have

$$\frac{\epsilon - \epsilon_2}{u_1} - \frac{\Delta_1 - \epsilon}{\Delta_1} = \mathbf{d}, \tag{B1}$$

$$\frac{\dfrac{\epsilon - \epsilon_2}{u_1} - \dfrac{\epsilon_2 - \epsilon_1}{u_1 + u_2}}{u_1} - \frac{\dfrac{\delta_2 - \delta_1}{\Delta_1 + \Delta_2} - \dfrac{\epsilon - \Delta_2}{\Delta_1}}{\Delta_1} = \mathbf{d}_2. \tag{B2}$$

Let $a_1 = 1$, $a_2 = \Delta_1/(u_1 + \Delta_1)$, $a_3 = u_1/(u_1 + \Delta_1)$ $a_4 = \Delta_1/(u_1 + u_2)$, $a_5 = 1 + (\Delta_1/(u_1 + u_2))$, $a_6 = 1 + (u_1/(\Delta_1 + \Delta_2))$, $a_7 = u_1/(\Delta_1 + \Delta_2)$, we can rewrite formulas (B1) and (B2) in matrix form as follows

$$\begin{pmatrix} a_1 & 0 & -a_2 & -a_3 & 0 \\ 0 & a_4 & -a_5 & a_6 & -a_7 \end{pmatrix} \cdot \begin{pmatrix} \epsilon \\ \epsilon_1 \\ \epsilon_2 \\ \delta_1 \\ \delta_2 \end{pmatrix} = \begin{pmatrix} \dfrac{u_1 \Delta_1}{u_1 + \Delta_1} \mathbf{d}_1 \\ u_1 \Delta_1 \mathbf{d}_2 + (u_1 - \Delta_1)\mathbf{d}_1 \end{pmatrix}, \tag{B3}$$

and

$$\begin{pmatrix} \epsilon \\ \epsilon_1 \\ \epsilon_2 \\ \delta_1 \\ \delta_2 \end{pmatrix} = \frac{1}{D} \cdot \begin{pmatrix} a_1(a_4^2 + a_5^2 + a_6^2 + a_7^2) & a_1(a_3 a_6 - a_2 a_5) \\ a_4(a_3 a_6 - a_2 a_5) & a_4(a_1^2 + a_2^2 + a_3^2) \\ -a_2(a_4^2 + a_5^2 + a_6^2 + a_7^2) - a_5(a_3 a_6 - a_2 a_5) & -a_2(a_3 a_6 - a_2 a_5) - a_5(a_1^2 + a_2^2 + a_3^2) \\ -a_3(a_4^2 + a_5^2 + a_6^2 + a_7^2) - a_6(a_3 a_6 - a_2 a_5) & -a_3(a_3 a_6 - a_2 a_5) - a_6(a_1^2 + a_2^2 + a_3^2) \\ -a_7(a_3 a_6 - a_2 a_5) & -a_7(a_1^2 + a_2^2 + a_3^2) \end{pmatrix} \cdot \begin{pmatrix} \dfrac{u_1 \Delta_1}{u_1 + \Delta_1} \mathbf{d}_1 \\ u_1 \Delta_1 \mathbf{d}_2 + (u_1 - \Delta_1)\mathbf{d}_1 \end{pmatrix}, \tag{B4}$$

where

$$D=(a_4^2+a_5^2+a_6^2+a_7^2)(a_1^2+a_2^2+a_3^2)-(a_3a_6-a_2a_5)^2.$$

By simple deduction, we have

$$\begin{aligned}&a_1(a_4^2+a_5^2+a_6^2+a_7^2)\\&>\max(|a_4(a_3a_6-a_2a_5)|,|-a_2(a_4^2+a_5^2+a_6^2+a_7^2)-a_5(a_3a_6\\&\quad-a_2a_5)|,|-a_3(a_4^2+a_5^2+a_6^2+a_7^2)-a_6(a_3a_6-a_2a_5)|,|\\&\quad-a_7(a_3a_6-a_2a_5)|),\end{aligned}$$

$$\frac{a_1(a_4^2+a_5^2+a_6^2+a_7^2)}{D}\le\frac{3}{4},$$

$$\frac{a_4(a_1^2+a_2^2+a_3^2)}{D}\le\frac{1}{2+\sqrt{2}},$$

$$\frac{a_7(a_1^2+a_2^2+a_3^2)}{D}\le\frac{1}{2+\sqrt{2}},$$

$$\frac{a_2(a_3a_6-a_2a_5)+a_5(a_1^2+a_2^2+a_3^2)}{D}\le\frac{1+a_2a_3+a_3^2}{3},$$

$$\frac{a_3(a_3a_6-a_2a_5)+a_6(a_1^2+a_2^2+a_3^2)}{D}\le\frac{1+a_2a_3+a_3^2}{3},$$

$$\left|\frac{a_1(a_3a_6-a_2a_5)}{D}\right|\le\frac{1}{3}$$

By substituting the above inequalities into Eqs. (B3) and (B4), we obtain Eqs. (16) and (17). This completes the proof. □

References

[1] Dannenberg L, Nowacki H. Approximate conversion of surface representations with polynomial bases. Comput-Aid Geometr Des 1985;2:123–31.

[2] Hoschek J. Approximate conversion of spline curves. Comput-Aid Geometr Des 1987;4:59–66.

[3] Watkins MA, Worsey AJ. Degree reduction of Bézier curves. Comput-Aid Des 1988;20:398–405.

[4] Eck M. Degree reduction of Bézier curves. Comput-Aid Geometr Des 1993;10:237–51.

[5] Eck M. Least square degree reduction of Bézier curves. Comput-Aid Des 1995;27:845–51.

[6] Bogacki P, Weinstein SE, Xu Y. Degree reduction of Bézier curves by uniform approximation with endpoint interpolation. Comput-Aid Des 1995;27:651–61.

[7] Hu SM, Jin TG. Approximate degree reduction of Bézier curves. Proceedings of Symposium on Computational Geometry, Hangzhou, China; 1992. p. 110–26 (also in: Tsinghua Sci Technol 3;1998:997–1000).

[8] Hu SM, Wang GZ, Jin TG. Properties of two types of generalized Ball curves. Comput-Aid Des 1996;28:218–22.

[9] Hu SM, Zheng GQ, Sun JG. Approximate degree reduction of rectangular Bezier surfaces. Chin J Software Res 1997;4:353–61.

[10] Hu SM, Zuo Z, Sun JG. Approximate degree reduction of triangular Bezier surfaces. Tsinghua Sci Technol 1998;3:1001–4.

[11] Piegl L. Algorithm for degree reduction of B-spline curves. Comput-Aid Des 1995;27:101–10.

[12] Yong JH, Hu SM, Sun JG, Tang XY. Degree reduction of B-spline curves. Comput Aid Geometr Des 2001;18:117–27.

[13] Wolters HJ, Wu G, Farin G. Degree reduction of B-spline curves. Computing 1998;13:235–41.

[14] Hu SM, Tong RF, Ju T, Sun JG. Approximate merging of a pair of Bézier curves. Comput-Aid Des 2001;33:125–36.

[15] Piegl L. The NURBS Book, 2nd ed. Berlin: Springer; 1997.

Chiew-Lan Tai has been an Assistant Professor at the Computer Science Department of the Hong Kong University of Science and Technology since August 1997. She received her BSc and MSc in Mathematics from the University of Malaya, and her MSc in Computer and Information Sciences from the National University of Singapore. She earned her DSc degree in Information Science from the University of Tokyo under a fellowship awarded by the Japan Society for the Promotion of Science. Her research interests includes geometric modeling, computer graphics, and graphics recognition.

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 post-doctor research at Tsinghua University in 1998. His current research interests are in computer-aided geometric design, computer graphics, and human-computer interaction.

Qi-Xing Huang is a graduate student in the Department of Computer Science and Technology, Tsinghua University, Beijing, People's Republic of China, where he received a BS in 2001. His research interests are computer aided geometric design and computer graphics.