COMPUTER-AIDED DESIGN

Computer-Aided Design 33 (2001) 125–136

www.elsevier.com/locate/cad

Approximate merging of a pair of Bézier curves

Shi-Min Hu[a,b,*], Rou-Feng Tong[c], Tao Ju[a,b], Jia-Guang Sun[a,b]

[a]National CAD Engineering Center, Tsinghua University, Beijing 100084, People's Republic of China
[b]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, People's Republic of China
[c]Department of Computer Science, Zhejiang University, Hangzhou 310027, People's Republic of China

Received 18 September 1999; received in revised form 24 April 2000; accepted 29 April 2000

Abstract

This paper deals with the merging problem, i.e. to approximate two adjacent Bézier curves by a single Bézier curve. A novel approach for approximate merging is introduced in the paper by using the constrained optimization method. The basic idea of this method is to find conditions for the precise merging of Bézier curves first, and then compute the constrained optimization solution by moving the control points. "Discrete" coefficient norm in L_2 sense and "squared difference integral" norm are used in our method. Continuity at the endpoints of curves are considered in the merging process, and approximate merging with points constraints are also discussed. Further, it is shown that the degree elevation of original Bézier curves will reduce the merging error.

Keywords: Bézier curve; Approximate conversion; Merging

1. Introduction and problem statement

Parametric polynomial representations are widely used in CAD systems which model free-form curves and surfaces. Bernstein–Bezier, Schoenberg-B-spline and Hermite–Coons type basis functions are frequently used in different systems. With the availability of a fast growing variety of modeling systems the demand has risen for exchanging curve and surface descriptions between various CAD systems [1]. The general aim when transferring geometric information from one system to another is to ensure a high degree of accuracy, the least possible loss of information and a small amount of geometric data for communication.

Conversion from one polynomial base to another can be achieved by direct matrix multiplication. For reducing the amount of communicating data, approximate conversion is considered by Hoschek [2]. As mentioned by Hoschek, the approximate conversion includes the following.

- *Degree reduction:* to find a parametric curve of degree n to approximate the given curve of degree $m (n < m)$.
- *Merging:* to merge as many as possible curve segments of degree M to one curve segment of degree $N (M \le N)$.

Degree reduction methods of Bézier and Ball curves have been extensively investigated by Watkins and Worsey [3], Lachance [4], Eck [5,6], Bogacki et al. [7] and Hu et al. [8,9] etc., degree reduction of B-spline curves has been considered by Piegl et al. [10], degree reduction of parametric surfaces also has been considered by Hu et al. [11,12]. However merging of Bézier curves is still an interesting and open problem.

Problem. For two adjacent Bézier curves $\mathbf{P}(u)$ and $\mathbf{Q}(v)$ $(0 \le u, v \le 1)$ with corresponding control points $\mathbf{P}_i$ and $\mathbf{Q}_i$ $(i = 0, 1, \ldots, n)$, merging of $\mathbf{P}(u)$ and $\mathbf{Q}(v)$ is a process that amounts to finding an n degree Bézier curve $\mathbf{R}(t)$ with control points $\mathbf{R}_i$ $(i = 0, 1, \ldots n)$, such that a suitable distance function $d(\mathbf{R}, \bar{\mathbf{R}})$ between $\mathbf{R}(t)$ and

$$\bar{\mathbf{R}}(t) = \begin{cases} \sum_{i=0}^{n} \mathbf{P}_i B_i^n\left(\dfrac{t}{\lambda}\right) & 0 \le t \le \lambda \\ \sum_{i=0}^{n} \mathbf{Q}_i B_i^n\left(\dfrac{t-\lambda}{1-\lambda}\right) & \lambda \le t \le 1 \end{cases} \tag{1}$$

is minimized on the interval [0, 1], where λ is a subdivision parameter.

The basic idea of this article is to find conditions for precise merging of Bézier curves. We modify control points of two Bézier curves such that the modified curves satisfy the precise merging conditions, then new control points of

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

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

0010-4485/01/$ - see front matter
PII: S0010-4485(00)00083-X

the merged curve can be obtained by an extrapolation algorithm.

The rest of the paper is organized as follows. Two constrained optimization methods with different optimization criteria are introduced, both of which achieve merging of Bézier curves by solving linear equation systems. An existence proof of the system of linear equations is given in Section 3. Conditions for matching original endpoints and derivatives are considered in Section 4, and merging with points constraints is dealt with in Section 5. Finally, we discuss approximate merging with degree elevation in Section 6.

2. Constrained optimization method for merging

2.1. Conditions of precise merging

Consider two n degree Bézier curves $\mathbf{P}(u)$ and $\mathbf{Q}(v)$, we first give conditions of precise merging. If $\mathbf{P}(u)$ and $\mathbf{Q}(v)$ can be merged precisely, this is to say, there exists an n degree Bézier curve $\mathbf{R}(t)=\sum_{i=0}^{n}\mathbf{R}_iB_i^n(t)$ such that $\mathbf{R}(t)=\bar{\mathbf{R}}(t)$ ($\bar{\mathbf{R}}(t)$ is defined as in Eq. (1)), so we have

$$\left.\frac{\partial^i\mathbf{P}(u)}{\partial u^i}\right|_{u=1}=\left.\frac{\partial^i\mathbf{Q}(v)}{\partial v^i}\right|_{v=0}\quad i=0,1,\dots n. \tag{2}$$

Note that

$$u=\frac{t}{\lambda},\ v=\frac{t-\lambda}{1-\lambda},$$

by the derivative formula of Bézier curves, Eq. (2) yields

$$\frac{1}{\lambda^i}\frac{n!}{(n-i)!}\Delta^i\mathbf{P}_{n-i}=\frac{1}{(1-\lambda)^i}\frac{n!}{(n-i)!}\Delta^i\mathbf{Q}_0, \tag{3}$$

where Δ is the difference operator defined as $\Delta P_i=P_{i+1}-P_i$, so we have the following theorem.

Theorem 1. *Two n degree Bézier curves* $\mathbf{P}(u)$ *and* $\mathbf{Q}(v)$ *can be merged precisely, if and only if there exists* $\mu(\mu>0)$, *such that*

$$\Delta^i\mathbf{P}_{n-i}=\mu^i\Delta^i\mathbf{Q}_0 \text{ for } i=0,1,\dots,n. \tag{4}$$

Proof.

1. *Necessity.* If $\mathbf{P}(u)$ and $\mathbf{Q}(v)$ can be merged precisely, from Eq. (3) we have

   $$\Delta^i\mathbf{P}_{n-i}=\mu^i\Delta^i\mathbf{Q}_0 \text{ for } i=0,1,\dots,n,$$

   where

   $$\mu=\frac{\lambda}{1-\lambda}.$$

   Therefore condition (4) is necessary.
2. *Sufficiency.* First, we represent Bézier curves in the exponential form. For any n degree Bézier curve $\mathbf{S}(t)$ with control points $\mathbf{S}_i$ $(i=0,1,\dots,n)$ we have

$$\mathbf{S}(t)=\sum_{i=0}^{n}t^i\binom{n}{i}\Delta^i\mathbf{S}_0=\sum_{i=0}^{n}(t-1)^i\binom{n}{i}\Delta^i\mathbf{S}_{n-i}. \tag{5}$$

Note that Eq. (5) holds true for any $t(t\ge 0)$. Next we construct a new Bézier curve $\mathbf{R}(t)$, whose control points $\mathbf{R}_j$ $(j=0,1,\dots,n)$ are defined by

$$\mathbf{R}_j=\sum_{i=0}^{j}\mathbf{P}_iB_i^j\left(\frac{1}{\lambda}\right),$$

where

$$\lambda=\frac{\mu}{\mu+1}$$

and $B_i^j(t)$ is the j degree Bézier base function. We will see that $\mathbf{P}(u)$ and $\mathbf{Q}(v)$ can be precisely merged into $\mathbf{R}(t)$. We calculate $\Delta^k\mathbf{R}_0(k=0,1,\dots n)$ as

$$\begin{aligned}\Delta^k\mathbf{R}_0&=\sum_{j=0}^{k}\left[\binom{k}{j}(-1)^{k-j}\sum_{i=0}^{j}\mathbf{P}_iB_i^j\left(\frac{1}{\lambda}\right)\right]\\&=\sum_{i=0}^{k}\left[\mathbf{P}_i\sum_{j=i}^{k}B_i^j\left(\frac{1}{\lambda}\right)\binom{k}{j}(-1)^{k-j}\right]\\&=\sum_{i=0}^{k}\mathbf{P}_i\binom{k}{i}(-1)^{k-i}\frac{1}{\lambda^k}=\frac{i}{\lambda^k}\Delta^k\mathbf{P}_0.\end{aligned} \tag{6}$$

Substitute Eq. (6) into exponential form (5), for $t(0\le t\le\lambda)$, we have

$$\mathbf{R}(t)=\sum_{i=0}^{n}t^i\binom{n}{i}\Delta^i\mathbf{R}_0=\sum_{i=0}^{n}\left(\frac{t}{\lambda}\right)^i\binom{n}{i}\Delta^i\mathbf{P}_0=\mathbf{P}\left(\frac{t}{\lambda}\right)$$

Given condition (4), for $t(\lambda\le t\le 1)$, we have

$$\begin{aligned}\mathbf{R}(t)&=\sum_{i=0}^{n}\left(\frac{t}{\lambda}\right)^i\binom{n}{i}\Delta^i\mathbf{P}_0\\&=\sum_{i=0}^{n}\left(\frac{t}{\lambda}-1\right)^i\binom{n}{i}\Delta^i\mathbf{P}_{n-i}\\&=\sum_{i=0}^{n}\left(\frac{t}{\lambda}-1\right)^i\mu^i\binom{n}{i}\Delta^i\mathbf{Q}_0=\mathbf{Q}\left(\frac{t-\lambda}{1-\lambda}\right)\end{aligned}$$

According to Eq. (1), $\mathbf{P}(u)$ and $\mathbf{Q}(v)$ can be precisely merged into $\mathbf{R}(t)$ with λ as the subdivision parameter. This completes the proof of Theorem 1.

□

2.2. Constrained optimization method (1)

We now consider the approximate merging problem stated in the previous Section 2.1. In order to obtain the merged Bézier curve, we will explore two different optimization criteria in the following discussion. First, we use the "discrete" coefficient norm, i.e. we set $d(\mathbf{E}(t), \mathbf{F}(t)) = \sum_{i=0}^{n} \|\mathbf{E}_i - \mathbf{F}_i\|^2$, where $\mathbf{E}(t)$ and $\mathbf{F}(t)$ are n degree Bézier curves with control points $\mathbf{E}_i$ and $\mathbf{F}_i$, $\|\cdot\|$ is the Euclidean norm. Minimization of the norm will result in the least difference between corresponding control points of the two curves. For Bézier curves $\mathbf{P}(u)$ and $\mathbf{Q}(v)$, perturbation $\epsilon_i = [\epsilon_i^x, \epsilon_i^y, \epsilon_i^z]^{\mathrm{T}}$ and $\delta_i = [\delta_i^x, \delta_i^y, \delta_i^z]^{\mathrm{T}}$ can be obtained for control points $\mathbf{P}_i$ of $\mathbf{P}(u)$ and $\mathbf{Q}_i$ of $\mathbf{Q}(v)$, respectively, such that the modified curves

$$\hat{\mathbf{P}}(u) = \sum_{i=0}^{n} \hat{\mathbf{P}}_i B_i^n(u) = \sum_{i=0}^{n} (\mathbf{P}_i + \epsilon_i) B_i^n(u) \;\; 0 \le u \le 1$$

$$\hat{\mathbf{Q}}(v) = \sum_{i=0}^{n} \hat{\mathbf{Q}}_i B_i^n(v) = \sum_{i=0}^{n} (\mathbf{Q}_i + \delta_i) B_i^n(v) \;\; 0 \le v \le 1$$

satisfies the conditions of Theorem 1, i.e., $\Delta^i \hat{\mathbf{P}}_{n-i} = \mu^i \Delta^i \hat{\mathbf{Q}}_0$. Then, $\hat{\mathbf{P}}(u)$and $\hat{\mathbf{Q}}(v)$ can be precisely merged into a Bézier curve $\mathbf{R}(t)$ of degree n, which is considered to be the approximate merged curve of $\mathbf{P}(u)$ and $\mathbf{Q}(v)$.

We determine ϵ_i, δ_i $(i = 0, 1, \ldots, n)$ by setting the optimization criterion $O(\epsilon_i, \delta_i)$ as

$$O(\epsilon_i, \delta_i) = \sum_{i=0}^{n} (\|\epsilon_i\|^2 + \|\delta_i\|^2) = \mathrm{Min}$$

and Lagrange function is defined by

$$L = \sum_{i=0}^{n} (\|\epsilon_i\|^2 + \|\delta_i\|^2) + \sum_{i=0}^{n} \lambda_i [\Delta^i (\mathbf{P}_{n-i} + \epsilon_{n-i}) - \mu^i \Delta^i (\mathbf{Q}_0 + \delta_0)], \tag{7}$$

where $\lambda_i = [\lambda_i^x, \lambda_i^y, \lambda_i^z]$ are Lagrange multipliers. In order to obtain an appropriate value of μ, we define a sequence $\mu_i (i = 1, 2, \ldots, n)$ such that

$$\Delta^i \mathbf{P}_{n-i} = \mu_i^i \Delta^i \mathbf{Q}_0$$

According to Theorem 1, the average value of the sequence will be a reasonable choice for parameter μ. Thus we take μ as

$$\mu = \frac{\left(\sum_{i=1}^{n} \mu_i\right)}{n} = \left(\frac{\sum_{i=1}^{n} \sqrt[i]{\frac{\|\Delta^i \mathbf{P}_{n-i}\|}{\|\Delta^i \mathbf{Q}_0\|}}}{n} \right). \tag{8}$$

Note that

$$\Delta^i \epsilon_{n-i} = \sum_{k=n-i}^{n} (-1)^{n-k} \binom{i}{n-k} \epsilon_k, \tag{9}$$

$$\Delta^i \delta_0 = \sum_{k=0}^{i} (-1)^{i-k} \binom{i}{k} \delta_k, \tag{10}$$

by setting

$$\frac{\partial L}{\partial \epsilon_j^x}, \frac{\partial L}{\partial \epsilon_j^y}, \frac{\partial L}{\partial \epsilon_j^z}, \frac{\partial L}{\partial \delta_j^x}, \frac{\partial L}{\partial \delta_j^y}, \frac{\partial L}{\partial \delta_j^z}, \frac{\partial L}{\partial \lambda_i^x}, \frac{\partial L}{\partial \lambda_i^y} \text{ and } \frac{\partial L}{\partial \lambda_i^z}$$

to be zero, we have

$$2\epsilon_j + \sum_{i=n-j}^{n} \lambda_i (-1)^{n-j} \binom{i}{n-j} = 0 \;\; j = 0, 1, \ldots, n, \tag{11}$$

$$2\delta_j - \sum_{i=j}^{n} \mu^i \lambda_i (-1)^{i-j} \binom{i}{j} = 0 \;\; j = 0, 1, \ldots, n, \tag{12}$$

$$\Delta^i \mathbf{P}_{n-i} + \Delta^i \epsilon_{n-i} - \mu^i \Delta^i \mathbf{Q}_0 - \mu^i \Delta^i \delta_0 = 0 \;\; i = 0, 1, \ldots, n. \tag{13}$$

Substituting Eqs. (11) and (12) into Eq. (13), produces the system of linear equations

$$\frac{1}{2} \sum_{l=0}^{n} \left[(1 + (-1)^{l+i} \mu^{l+i}) \sum_{k=0}^{\min(l,i)} \binom{i}{k} \binom{l}{k} \right] \lambda_l = \Delta^i \mathbf{P}_{n-i} - \mu^i \Delta^i \mathbf{Q}_0 \;\; i = 0, 1, \ldots, n. \tag{14}$$

Eq. (14) can be further simplified as

$$\frac{1}{2} \sum_{l=0}^{n} \left[(1 + (-\mu)^{l+i}) \binom{l+i}{i} \right] \lambda_l = \Delta^i \mathbf{P}_{n-i} - \mu^i \Delta^i \mathbf{Q}_0 \quad i = 0, 1, \ldots, n. \tag{15}$$

So λ_l $(l = 0, 1, \ldots, n)$ can be obtained by solving the above system of linear equations, and ϵ_i, $\delta_i (i = 0, 1, \ldots, n)$ can be obtained from Eqs. (11) and (12). The existing proof of solution of the linear equation system (15) will be given in Section 3.

2.3. Constrained optimization method (2)

We can also use the "squared difference integral" norm, such that

$$d(\mathbf{E}(t), \mathbf{F}(t)) = \int_0^1 (\mathbf{E}(t) - \mathbf{F}(t))^2 \, \mathrm{d}t,$$

where $\mathbf{E}(t)$ and $\mathbf{F}(t)$ are n degree Bézier curves. The norm can be viewed geometrically as the squared area between two Bézier curves. Therefore, minimization of the norm will result in the least enclosing area between the two curves.

We determine perturbations ϵ_i, $\delta_i (i = 0, 1, \ldots, n)$, defined in Section 2.2, by setting the optimization criterion $O(\epsilon_i, \delta_i)$ as

$$O(\epsilon_i, \delta_i) = \int_0^1 \left[\left(\sum_{i=0}^{n} B_i^n(t) \epsilon_i \right)^2 + \left(\sum_{i=0}^{n} B_i^n(t) \delta_i \right)^2 \right] \mathrm{d}t = \mathrm{Min}$$

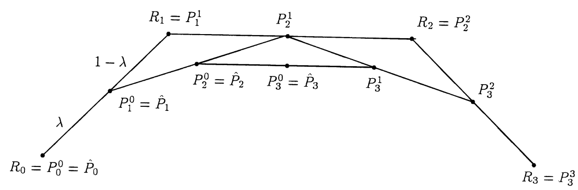

Fig. 1. Compute new control points by extrapolation.

and Lagrange function is defined by

$$L = \int_0^1 \left[\left(\sum_{i=0}^{n} B_i^n(t)\epsilon_i \right)^2 + \left(\sum_{i=0}^{n} B_i^n(t)\delta_i \right)^2 \right] \mathrm{d}t$$

$$+ \sum_{i=0}^{n} \lambda_i [\Delta^i(\mathbf{P}_{n-i} + \epsilon_{n-i}) - \mu^i \Delta^i(\mathbf{Q}_0 + \delta_0)] \quad (16)$$

where $\lambda_i = [\lambda_i^x, \lambda_i^y, \lambda_i^z]$ are Lagrange multipliers and μ is defined as in Section 2.2. By setting

$$\frac{\partial L}{\partial \epsilon_j^x}, \frac{\partial L}{\partial \epsilon_j^y}, \frac{\partial L}{\partial \epsilon_j^z}, \frac{\partial L}{\partial \delta_j^x}, \frac{\partial L}{\partial \delta_j^y}, \frac{\partial L}{\partial \delta_j^z}, \frac{\partial L}{\partial \lambda_i^x}, \frac{\partial L}{\partial \lambda_i^y} \text{ and } \frac{\partial L}{\partial \lambda_i^z}$$

to be zero, we have

$$2\sum_{i=0}^{n} \epsilon_i N_{ij} + \sum_{i=n-j}^{n} \lambda_i (-1)^{n-j} \binom{i}{n-j} = 0 \quad j = 0, 1, \ldots, n, \quad (17)$$

$$2\sum_{i=0}^{n} \delta_i N_{ij} - \sum_{i=j}^{n} \mu^i \lambda_i (-1)^{i-j} \binom{i}{j} = 0 \quad j = 0, 1, \ldots, n, \quad (18)$$

$$\Delta^i \mathbf{P}_{n-i} + \Delta^i \epsilon_{n-i} - \mu^i \Delta^i \mathbf{Q}_0 - \mu^i \Delta^i \delta_0 = 0 \quad i = 0, 1, \ldots, n. \quad (19)$$

where N_{ij} is defined as

$$N_{ij} = \int_0^1 B_i^n(t) B_j^n(t)\, \mathrm{d}t \quad i, j = 0, 1, \ldots, n. \quad (20)$$

Thus ϵ_i, δ_i and $\lambda_i (i = 0, 1, \ldots, n)$ can be obtained by solving the linear equation system consisting of Eqs. (17)–(19). The existence proof will be given in Section 3.

2.4. Calculation of new control points $\boldsymbol{R}_i$

We are now in a position to give a recursive algorithm to compute new control points of the merged curve $\mathbf{R}(t)$. Since control points of $\hat{\mathbf{P}}(u)$ and $\hat{\mathbf{Q}}(v)$ can be derived by subdividing $\mathbf{R}(t)$ at $t = \lambda = (\mu/(1+\mu))$ (see Fig. 1), we have the following algorithm to compute new control points by using extrapolation.

Algorithm

Set $\lambda = \mu/(1+\mu)$
Set $\mathbf{P}_i^0 = \mathbf{P}_i + \epsilon_i$ for $i = 0, 1, \ldots, n$
For $j = 1, 2, \ldots, n$
For $i = j, j+1, \ldots, n$
$\mathbf{P}_i^j = (1 - 1/\lambda)\mathbf{P}_{i-1}^{j-1} + 1/\lambda \mathbf{P}_i^{j-1}$
Next i
Next j
$\mathbf{R}_k = \mathbf{P}_k^k$ for $k = 0, 1, \ldots, n$.

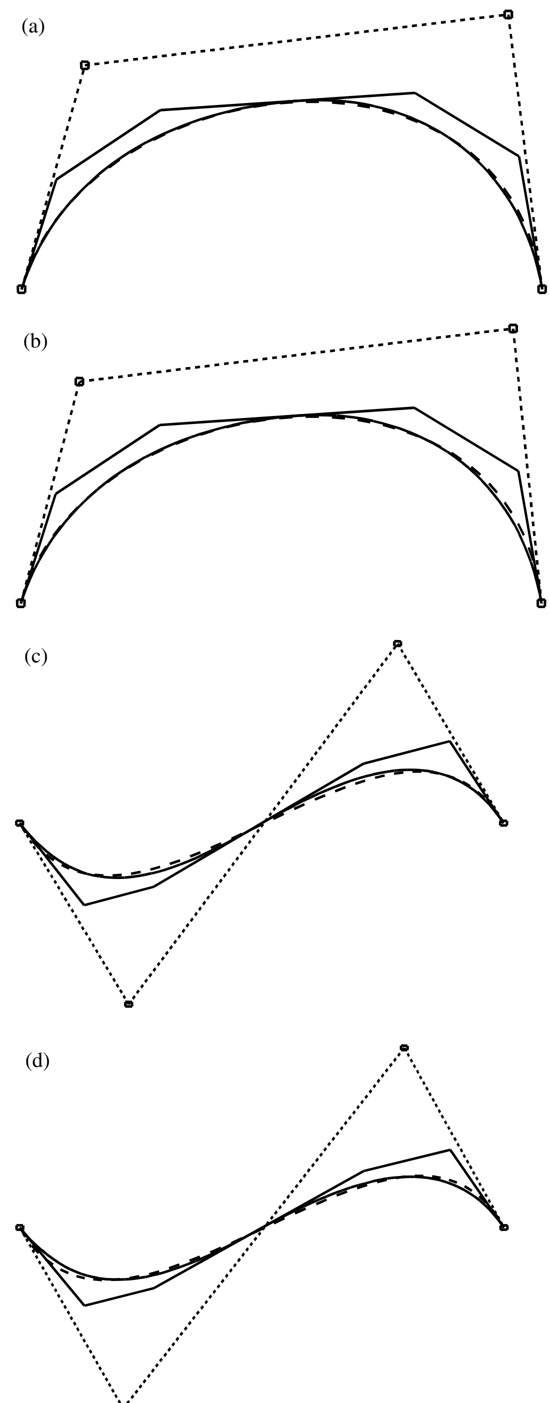

Fig. 2. Merging of two cubic Bézier curves.

Examples. In Fig. 2, we demonstrate merging of cubic Bézier curves pair using the constrained optimization methods discussed so far. Examples (a) and (c) adopt the first constrained optimization method, while (b) and (d) use the second optimization method in merging. The original curves and control polygons are shown as solid lines, while the merged curves and associated polygons are shown as dashed lines. We can see that the first method works better in (a), while the second method achieves nicer results in (d).

3. An existence proof of solution

In this section, we first give the existence proof of system (15). By rewriting Eqs. (7) and (13) in the matrix form, we get

$$L = \eta^{\mathrm{T}}\cdot\eta + \lambda^{\mathrm{T}}(A\eta + B) \tag{21}$$

and

$$A\eta + B = 0 \tag{22}$$

in which $\eta=(\epsilon_0, \epsilon_1, \ldots, \epsilon_n, \delta_0, \delta_1, \ldots, \delta_n)^{\mathrm{T}}, \lambda = (\lambda_0, \lambda_1, \ldots, \lambda_n)^{\mathrm{T}}$,

$$A = (a_{ij})_{(n+1)\times(2n+2)}$$

$$B = [b_0, b_1, \ldots, b_n]^{\mathrm{T}},$$

where

$$a_{ij} = \begin{cases} 0, & \text{for } 0 \le i + j < n \text{ or } n + i + 1 < j \le 2n + 1 \\ (-1)^{n-j}\begin{pmatrix} i \\ n-j \end{pmatrix}, & \text{for } 0 \le i,j \le n,\ i + j \ge n \\ (-1)^{n+i-j}\mu^i \begin{pmatrix} i \\ j-n-1 \end{pmatrix}, & \text{for } 0 \le i \le n,\ n+1 \le j \le n+i+1 \end{cases}$$

$$b_i = \Delta^i \mathbf{P}_{n-i} - \mu^i \Delta^i \mathbf{Q}_0.$$

Differentiating Eq. (21) with respect to η, we obtain (Eqs. (11) and (12) in the matrix form)

$$2\eta + A^{\mathrm{T}}\lambda = 0 \tag{23}$$

By substituting Eq. (23) into Eq. (22), we obtain (Eq. (15) in matrix form)

$$\tfrac{1}{2}AA^{\mathrm{T}}\lambda = B. \tag{24}$$

Note that

$$A = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 & 0 & * & * & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & * & * & * & * & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ldots & * & * & * & * & * & * & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & * & \ldots & * & * & * & * & * & * & \ldots & * & 0 & 0 \\ 0 & * & * & \ldots & * & * & * & * & * & * & \ldots & * & * & 0 \\ * & * & * & \ldots & * & * & * & * & * & * & \ldots & * & * & * \end{pmatrix}_{(n+1)\times(2n+2)}$$

is obviously a row-full-rank matrix. So $\frac{1}{2}AA^{\mathrm{T}}$ is a positive definite matrix, thus it's non-singular. This completes the existence proof of the system (15).

Similarly, by rewriting Eq. (16) in matrix form, we get

$$L = \eta^{\mathrm{T}}C\eta + \lambda^{\mathrm{T}}(A\eta + B). \tag{25}$$

here C is a symmetric matrix defined as $C = (c_{ij})_{(2n+2)\times(2n+2)}$ in which

$$c_{ij} = \begin{cases} 0, & \text{for } 0 \le i \le n \text{ and } n < j \le 2n+1, \\ & \text{or } 0 \le j \le n \text{ and } n < i \le 2n+1 \\ N_{ij}, & \text{for } 0 \le i,j \le n \\ N_{(i-n-1)(j-n-1)}, & \text{for } n < i,j \le 2n+1 \end{cases},$$

where N_{ij} are defined in Eq. (20). According to Eq. (16), $\eta^T C\eta$ is always positive for any $\eta(\eta \ne 0)$, therefore C is positive definite. Differentiating Eq. (25) with respect to η, we get (Eqs. (17) and (18) in matrix form)

$$2C\eta + A^{\mathrm{T}}\lambda = 0. \tag{26}$$

Eq. (19) can also be written in matrix form

$$A\eta + B = 0. \tag{27}$$

Substituting Eq. (26) into Eq. (27), we obtain

$$\tfrac{1}{2}AC^{-1}A^{\mathrm{T}}\lambda = B \tag{28}$$

because A is a row-full-rank matrix and C^{-1} is symmetric positive definite, so $\frac{1}{2}AC^{-1}A^{\mathrm{T}}$ is non-singular [13]. By substituting Eq. (28) into Eq. (26), we obtain η as

$$\eta + C^{-1}A^{\mathrm{T}}(AC^{-1}A^{\mathrm{T}})^{-1}B = 0$$

This completes the existence proof of the linear equation system consisting of Eqs. (17)–(19).

4. Conditions for matching original endpoints and derivatives

To match original endpoints $\mathbf{P}_0$ and $\mathbf{Q}_n$, two constraints $\epsilon_0 = 0,\ \delta_n = 0$ should be added. Take the first constrained optimization method for example, by discussion similar to

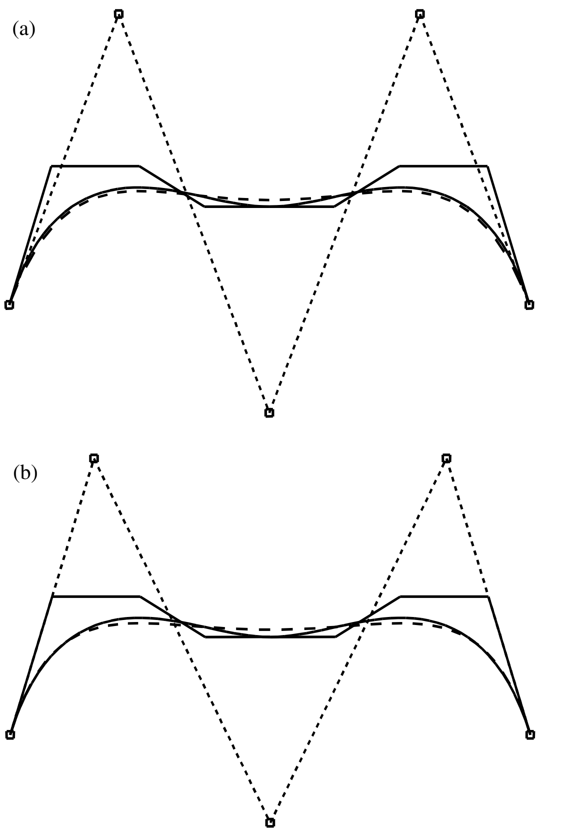

Fig. 3. Merging of two quartic Bézier curves with endpoints matching.

that in Section 2.2, we obtain a system of linear equations

$$\begin{cases} \frac{1}{2}\sum_{l=0}^{n}\left[(1+(-\mu)^{l+i})\binom{l+i}{i}\right]\lambda_l = \Delta^i\mathbf{P}_{n-i} - \mu^i\Delta^i\mathbf{Q}_0 \\ i=0,1,\ldots,n-1 \\ \frac{1}{2}\sum_{l=0}^{n}\left[(1+(-\mu)^{l+n})\sum_{k=0}^{\min(l,n-1)}\binom{n}{k}\binom{l}{k}\right]\lambda_l = \Delta^n\mathbf{P}_0 - \mu^n\Delta^n\mathbf{Q}_0 \end{cases}$$

Then, λ_i $(i=0,1,\ldots,n)$ can be obtained, and ϵ_j, $\delta_j(j=0,1,\ldots,n)$ can be obtained as follows

$$\begin{cases} \epsilon_0 = 0 \\ \epsilon_j = -\frac{1}{2}\sum_{i=n-j}^{n}(-1)^{n-j}\binom{i}{n-j}\lambda_i \quad j=1,\ldots,n \\ \delta_j = \frac{1}{2}\sum_{i=j}^{n}(-1)^{i-j}\binom{i}{j}\mu^i\lambda_i \qquad j=0,1,\ldots,n-1 \\ \delta_n = 0 \end{cases}$$

Further, constraints $\epsilon_1 = 0$ and $\delta_{n-1} = 0$ could be added to the above equation system so that the merged curve would match the original derivatives at the left and right ends of the pair of Bézier curves. The system of linear equations would become

$$\begin{cases} \frac{1}{2}\sum_{l=0}^{n}\left[(1+(-\mu)^{l+i})\binom{l+i}{i}\right]\lambda_l = \Delta^i\mathbf{P}_{n-i} - \mu^i\Delta^i\mathbf{Q}_0 \\ i=0,1,\ldots,n-2 \\ \frac{1}{2}\sum_{l=0}^{n}\left[(1+(-\mu)^{l+j})\sum_{k=0}^{\min(l,n-2)}\binom{j}{k}\binom{l}{k}\right]\lambda_l = \Delta^j\mathbf{P}_0 - \mu^j\Delta^j\mathbf{Q}_0 \\ j=n-1,n \end{cases}$$

Then ϵ_j, δ_j $(j=0,1,\ldots,n)$ can be obtained as follows

$$\begin{cases} \epsilon_0 = \epsilon_1 = 0 \\ \epsilon_j = -\frac{1}{2}\sum_{i=n-j}^{n}(-1)^{n-j}\binom{i}{n-j}\lambda_i \quad j=2,\ldots,n \\ \delta_j = \frac{1}{2}\sum_{i=j}^{n}(-1)^{i-j}\binom{i}{j}\mu^i\lambda_i \qquad j=0,1,\ldots,n-2 \\ \delta_n = \delta_{n-1} = 0 \end{cases}$$

Similarly, for the second constrained optimization method, matching of original endpoints and derivatives can be achieved by adding constraints $\epsilon_0 = \epsilon_1 = 0$ and $\delta_n = \delta_{n-1} = 0$ to Lagrange function (16). Perturbations ϵ_i, δ_i can then be obtained by solving the corresponding system of linear equations.

Examples. In Fig. 3, we illustrate merging results of two quartic Bézier curves using the first optimization method. Endpoints matching is implemented in (a), and matching of derivatives at endpoints is achieved in (b). The original curves and control polygons are shown as solid lines, and the merged curves and associated polygons are shown as dashed lines.

5. Approximate merging with points constraints

A more general problem of matching endpoints is for the merged curve to pass through target points, not only endpoints on the original Bézier curves. This can be achieved by adding points constrained conditions to the Lagrange function. Given that

$$\sum_{i=0}^{n} B_i^n(u_j)\mathbf{P}_i = \sum_{i=0}^{n} B_i^n(u_j)(\mathbf{P}_i + \epsilon_i) \;\; j=0,1,\ldots,l,$$

$$\sum_{i=0}^{n} B_i^n(v_k)\mathbf{Q}_i = \sum_{i=0}^{n} B_i^n(v_k)(\mathbf{Q}_i + \delta_i) \;\; k=0,1,\ldots,m,$$

where $u_j(j=0,1,\ldots,l)$ and $v_k(j=0,1,\ldots,m)$ are parameters of different target points on $\mathbf{P}(u)$ and $\mathbf{Q}(v)$, respectively, we have

$$\sum_{i=0}^{n} B_i^n(u_j)\epsilon_i = 0 \;\; j=0,1,\ldots,l, \tag{29}$$

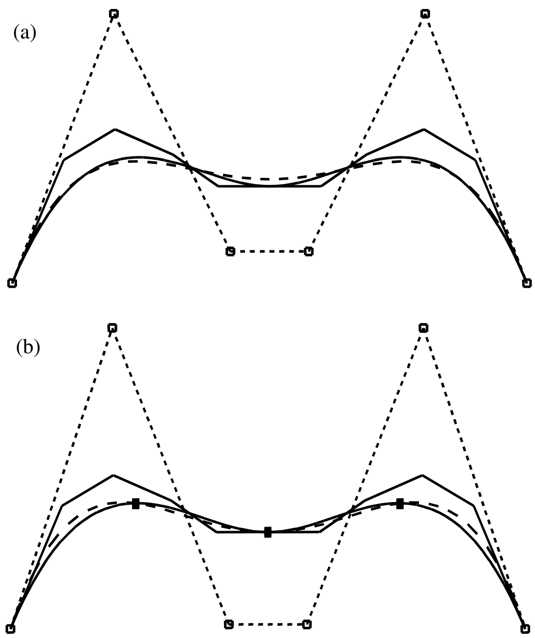

Fig. 4. Merging of two quintic Bézier curves with points constraints.

$$\sum_{i=0}^{n} B_i^n(v_k)\delta_i = 0 \quad k = 0, 1, \ldots, m. \tag{30}$$

By adding the above-constrained conditions, we rewrite the Lagrange function as

$$L = O(\epsilon_i, \delta_i) + \sum_{i=0}^{n} \lambda_i[\Delta^i(\mathbf{P}_{n-i} + \epsilon_{n-i}) - \mu^i \Delta^i(\mathbf{Q}_0 + \delta_0)] + \sum_{j=0}^{l} \lambda_{n+j+1}\left(\sum_{i=0}^{n} B_i^n(u_j)\epsilon_i\right) + \sum_{k=0}^{m} \lambda_{n+l+k+2}\left(\sum_{i=0}^{n} B_i^n(v_k)\delta_i\right), \tag{31}$$

where $O(\epsilon_i, \delta_i)$ is the appropriate optimization criterion. Differentiating Eq. (31) with respect to ϵ_i, $\delta_i (i = 0, 1, \ldots, n)$ and $\lambda_i (i = 0, 1, \ldots, n + l + m + 2)$, using similar techniques described in Sections 2.2 and 2.3, we obtain a linear system of $(3n + m + l + 5)$ equations consisting of Eqs. (13), (29), (30) and

$$\frac{\partial O(\epsilon_i, \delta_i)}{\partial \epsilon_i} + \sum_{j=n-i}^{n} \lambda_i (-1)^{n-i} \binom{j}{n-i} + \sum_{j=0}^{l} \lambda_{n+j+1} B_i^n(u_j) = 0 \qquad i = 0, 1, \ldots, n, \tag{32}$$

$$\frac{\partial O(\epsilon_i, \delta_i)}{\partial \delta_i} - \sum_{j=i}^{n} \mu^j \lambda_i (-1)^{j-i} \binom{j}{i} + \sum_{j=0}^{m} \lambda_{n+j+l+2} B_i^n(v_j) = 0 \qquad i = 0, 1, \ldots, n. \tag{33}$$

Note that unlike the equation system in Section 2.3, the above linear system is not solvable when the total number of constrained conditions in Eqs. (13), (29) and (30) exceeds the number of constrained variables (ϵ_i and δ_i). Generally speaking, the total number of different points constraints on $\mathbf{P}(u)$ and $\mathbf{Q}(v)$ $(l + m + 2)$ should not be larger than $(n + 1)$ for a solvable solution.

Example. Points constraints can be used in preserving the geometric details of the original pair of Bézier curves. In Fig. 4(a), two quintic Bézier curves are merged using the second optimization method with endpoints matching, i.e. $u_0 = 0.0$, $v_0 = 1.0$, while (b) illustrates the merging results after three more points constraints at $u_0 = 0.5$, $u_1 = 1.0$, $v_0 = 0.5$ are added.

6. Constrained optimization with degree elevation

In this section, we will discuss constrained optimization with degree elevation. Sometimes, two adjacent Bézier curves are not suitable to be approximated by a new Bézier curve with the same degree. For example, a cubic Bézier curve can have at most one inflection point, so merging a pair of Bézier curves that each has an inflection point into a new cubic Bézier curve will not be a good idea. A reasonable choice in this case is to merge two cubic curves into a curve of degree 4 or 5. In effect, better approximation can be achieved when two n degree Bézier curves $\mathbf{P}(u)$ and $\mathbf{Q}(v)$ are merged into a new Bézier curve after the degree elevation.

We can think of the increase in approximation precision in this way. Constrained optimization can be viewed as a two-step process: firstly, a set of Bézier curve pairs $\hat{\mathbf{P}}(u)$ and $\hat{\mathbf{Q}}(v)$ are obtained such that each pair can be precisely merged into a Bézier curve; secondly, the curves pair with the least optimization norm is "chosen" and merged as the final approximate merged curve. Since each pair of $\hat{\mathbf{P}}(u)$ and $\hat{\mathbf{Q}}(v)$ obtained in the first step can be elevated to a higher degree and still satisfy the condition of precise merging, the set of curve pairs obtained in the first step of constrained optimization of degree n is only a subset of those obtained in the first step for degree $n + 1$ or above. In other words, optimization method will have more eligible curve pairs to "choose from" after degree elevation. Consequently better approximate merged curve will be obtained.

We illustrate this process by using the second optimization method. First, we prove a theorem about degree elevation.

Theorem 2. *$\mathbf{P}(u)$ is a n degree Bézier curve with control points $\mathbf{P}_i (i = 0, 1, \ldots, n)$, and Δ is the difference operator, we have*

$$\Delta^i \mathbf{P}^*_{n+1-i} = \left(1 - \frac{i}{n+1}\right)\Delta^i \mathbf{P}_{n-i} \quad i = 0, 1, \ldots, n + 1, \tag{34}$$

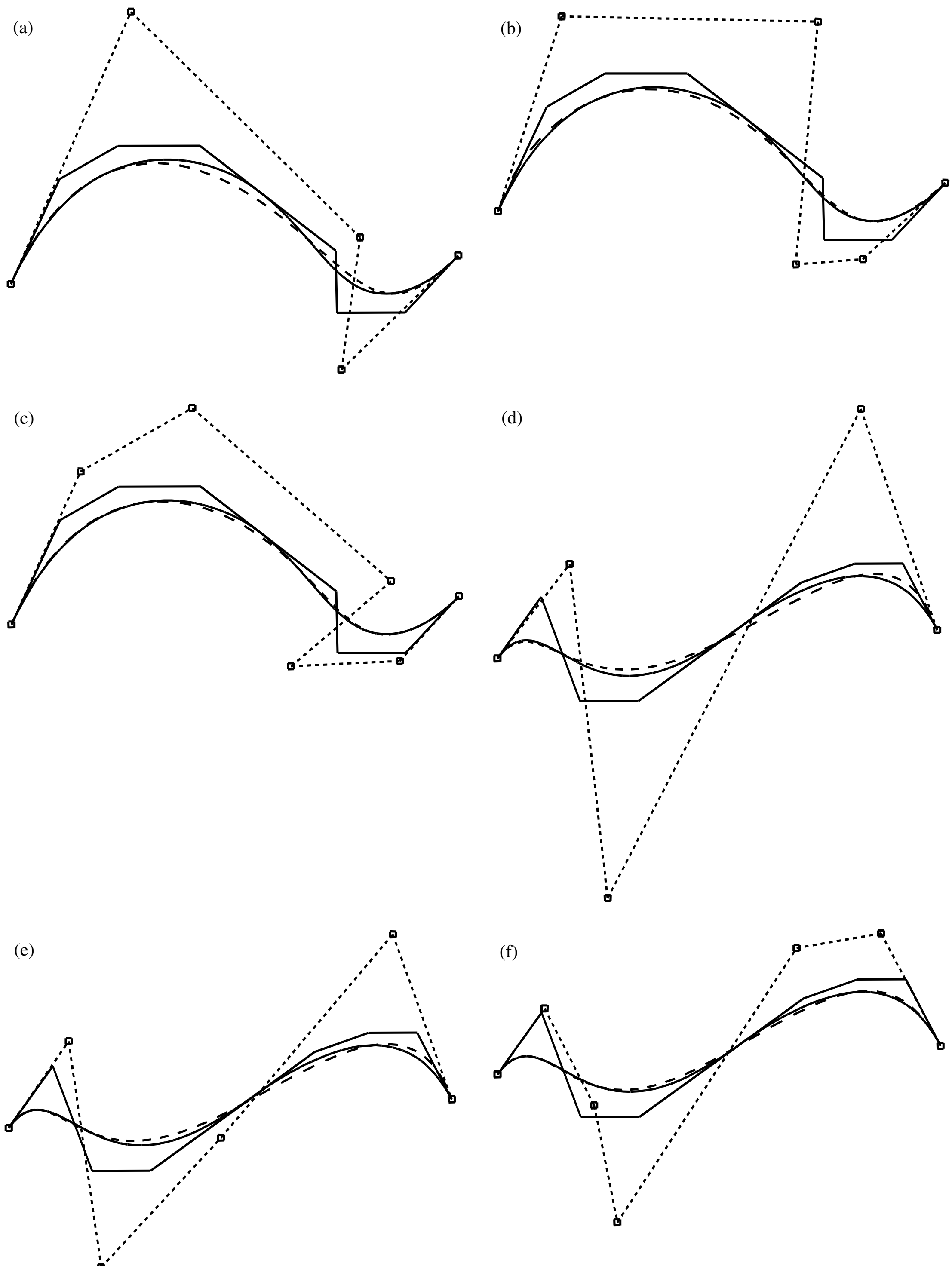

Fig. 5. Merging of two quartic Bézier curves with degree elevation.

$$\Delta^i \mathbf{P}_0^* = \left(1 - \frac{i}{n+1}\right)\Delta^i \mathbf{P}_0 \quad i = 0, 1, \ldots, n+1 \tag{35}$$

where $\mathbf{P}_i^*(\, i = 0, 1, \ldots, n+1)$ *are* $n+1$ *degree control points after degree elevation of* $\mathbf{P}(u)$.

Proof. Note that

$$\mathbf{P}_i^* = \left(1 - \frac{i}{n+1}\right)\mathbf{P}_i + \frac{i}{n+1}\mathbf{P}_{i-1} \quad i = 0, 1, \ldots, n+1, \tag{36}$$

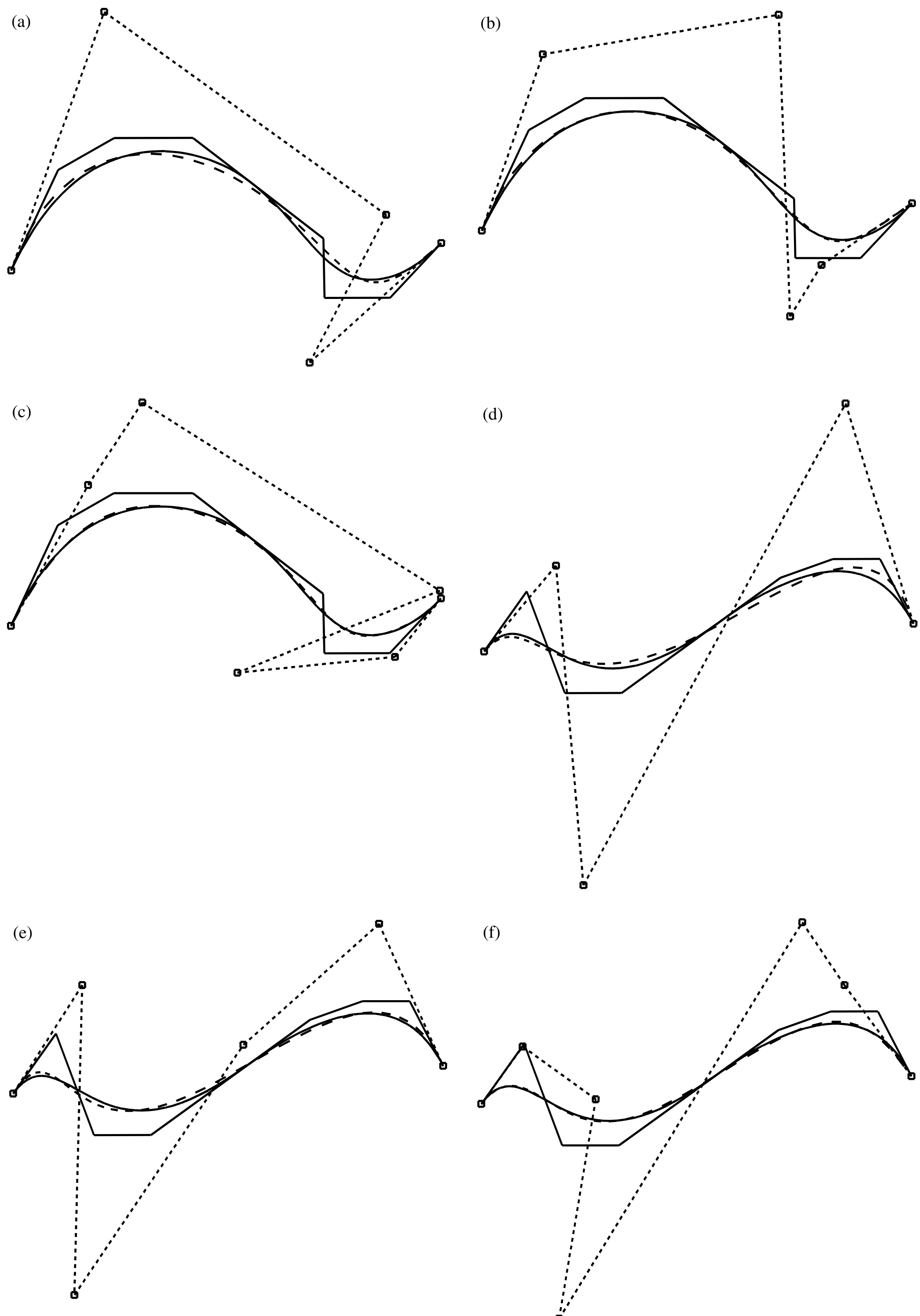

Fig. 6. Merging of two quartic Bézier curves with degree elevation.

Table 1
Comparison of merging error after degree elevation

	μ	$\mathbf{E}(\mathbf{R}(t),\mu)$		
		Degree 4	Degree 5	Degree 6
Example (a)	0.934426	1.051789	0.498082	0.230428
Example (d)	0.753846	1.144209	0.403908	0.192052

and

$$\Delta^i \mathbf{P}^*_{n+1-i} = \sum_{k=0}^{i} (-1)^{i-k} \binom{i}{k} \mathbf{P}^*_{n+1+k-i} \quad i = 0, 1, \ldots, n+1, \tag{37}$$

$$\Delta^i \mathbf{P}^*_0 = \sum_{k=0}^{i} (-1)^k \binom{i}{k} \mathbf{P}^*_{i-k} \quad i = 0, 1, \ldots, n+1. \tag{38}$$

Substituting Eq. (36) into Eqs. (37) and (38) would yield Eqs. (34) and (35). This completes the proof of Theorem 2.

Theorem 2 guarantees that the value of μ (defined in Eq. (8)) does not change after degree elevation. Thus merging methods discussed in this paper will proceed with the same parameter μ for Bézier curves pair after degree elevation.

Given two adjacent Bézier curves of degree n, $\mathbf{P}(u)$ and $\mathbf{Q}(v)$, we first consider the approximate merged curve $\mathbf{R}(t)$ using the second optimization method. Here, we define the merging error $\mathbf{E}$ as in Section 2.3, such that

$$\mathbf{E}(\mathbf{R}(t),\mu) = \int_0^1 (\hat{\mathbf{P}}(u) - \mathbf{P}(u))^2 \,\mathrm{d}u + \int_0^1 (\hat{\mathbf{Q}}(v) - \mathbf{Q}(v))^2 \,\mathrm{d}v, \tag{39}$$

where $\hat{\mathbf{P}}(u)$ and $\hat{\mathbf{Q}}(v)$ can be precisely merged into $\mathbf{R}(t)$ with subdivision parameter $\mu/(1+\mu)$.

After the degree elevation of Bézier curves $\mathbf{P}(u)$ and $\mathbf{Q}(v)$, we obtain two new Bézier curves $\mathbf{P}^*(u)$ and $\mathbf{Q}^*(v)$ of degree $n+1$. Similarly, we obtain $n+1$ degree Bézier curves $\hat{\mathbf{P}}^*(u)$ and $\hat{\mathbf{Q}}^*(v)$ after degree elevation of $\hat{\mathbf{P}}(u)$ and $\hat{\mathbf{Q}}(v)$. Therefore Bézier curves $\hat{\mathbf{P}}^*(u)$ and $\hat{\mathbf{Q}}^*(v)$ can also be precisely merged into $\mathbf{R}^*(t)$, which is the $n+1$ degree Bézier curve elevated from $\mathbf{R}(t)$. Examine the merging error of $\mathbf{R}*(t)$ with Eq. (39), we have

$$\mathbf{E}(\mathbf{R}^*(t),\mu)$$
$$= \int_0^1 (\hat{\mathbf{P}}^*(u) - \mathbf{P}^*(u))^2 \,\mathrm{d}u + \int_0^1 (\hat{\mathbf{Q}}^*(v) - \mathbf{Q}^*(v))^2 \,\mathrm{d}v$$
$$= \int_0^1 (\hat{\mathbf{P}}(u) - \mathbf{P}(u))^2 \,\mathrm{d}u + \int_0^1 (\hat{\mathbf{Q}}(v) - \mathbf{Q}(v))^2 \,\mathrm{d}v$$
$$= \mathbf{E}(\mathbf{R}(t),\mu)$$

On the other hand, for $n+1$ degree Bézier curves $\mathbf{P}^*(u)$ and $\mathbf{Q}*(v)$, we compute an approximate merged curve $\bar{\mathbf{R}}(t)$ of degree $n+1$ with parameter μ using the second optimization method. According to the optimization criterion, we have

$$\mathbf{E}(\bar{\mathbf{R}}(t),\mu) = \mathrm{Min} \le \mathbf{E}(\mathbf{R}^*(t),\mu) = \mathbf{E}(\mathbf{R}(t),\mu) \tag{40}$$

From Eq. (40) we can conclude that $\bar{\mathbf{R}}(t)$ results in less merging error $\mathbf{E}$ than its n degree counterpart $\mathbf{R}(t)$, and thus achieves better approximation after degree elevation. Similar results can be obtained for the first optimization method, with the definition of merging error $\mathbf{E}$ as

$$\mathbf{E}(\mathbf{R}(t),\mu) = \sum_{i=0}^{n} (\|\epsilon_i\|^2 + \|\delta_i\|^2),$$

where ϵ_i and δ_i are perturbations of control points of $\hat{\mathbf{P}}(u)$ and $\hat{\mathbf{Q}}(v)$ from $\mathbf{P}(u)$ and $\mathbf{Q}(v)$. Practical examples of merging with both methods are provided in Figs. 5 and 6.

Examples. Two merging examples of quartic cases are presented in Fig. 5 by using the first optimization method. Two original quartic Bézier curves are merged into a new quartic Bézier curve each in (a) and (d), and are then merged into Bézier curves of degree 5 and 6, respectively, in (b), (c) and (e), (f) after degree elevation. The original curves and control polygons are shown as solid lines, and the merged curves and associated polygons are shown as dashed lines.

Table 1 contains the comparison results of merging error of each example presented in Fig. 5. From Table 1 we can see that the merging error $\mathbf{E}(\mathbf{R}(t),\mu)$ decreases each time after degree elevation.

Examples. Two merging examples of quartic cases are presented in Fig. 6 by using the second optimization method. Two original quartic Bézier curves are merged into a new quartic Bézier curve each in (a) and (d), and are then merged into Bézier curves of degree 5 and 6, respectively, in (b), (c) and (e), (f) after degree elevation. The original curves and control polygons are shown as solid lines, and the merged curves and associated polygons are shown as dashed lines.

Table 2 contains results of merging error of each example

Table 2
Comparison of merging error after degree elevation

	μ	$\mathbf{E}(\mathbf{R}(t),\mu)$		
		Degree 4	Degree 5	Degree 6
Example (a)	0.934426	0.023273	0.008274	0.003361
Example (d)	0.753846	0.020258	0.007306	0.002114

presented in Fig. 6. From Table 2 we can see that merging error $\mathbf{E}(\mathbf{R}(t),\mu)$ decreases each time after degree elevation.

7. Remarks

Remark 1. *In general, the discussion in this paper can apply to all pairs of curves with G^0 joining. However, Theorem 1 prescribes that the merged curve should have at least G^1 continuity at the connecting point. Therefore we recommend using the methods presented in the paper to pairs of Bézier curves with at least G^1 joining in order to avoid discrepancy at connecting point.*

Remark 2. *Strictly speaking, the subdivision parameter μ in Eq. (7) should be dependent on $\epsilon_i, \delta_i (i = 0, 1, \ldots, n)$, such that*

$$\mu = \frac{\|\mathbf{P}_n + \epsilon_n - \mathbf{P}_{n-1} - \epsilon_{n-1}\|}{\|\mathbf{Q}_1 + \delta_1 - \mathbf{Q}_0 - \delta_0\|}.$$

Under such a choice, the Lagrange function L will get too complicated. So we take μ as in Eq. (8), where μ is independent on $\epsilon_i, \delta_i (i = 0, 1, \ldots, n)$. However, a better choice μ' can be obtained from μ so that the merging error will get smaller. By using μ defined in Eq. (8), the merged curve $\mathbf{R}(t)$ of $\mathbf{P}(u)$ and $\mathbf{Q}(v)$ can be computed firstly as in Section 2. We hope to obtain better parameter μ'. With the unknown parameter μ', we subdivide the merged curve $\mathbf{R}(t)$ into two Bézier curves with control points $\mathbf{S}'_i$ and $\mathbf{T}'_i$ $(i = 0, 1, \ldots, n)$, and define $\epsilon'_i, \delta'_i (i = 0, 1, \ldots, n)$ as

$$\epsilon'_i = \mathbf{S}'_i - \mathbf{P}_i = \sum_{j=0}^{i} B^i_j(\mu')\mathbf{R}_j - \mathbf{P}_i \quad i = 0, 1, \ldots, n \tag{41}$$

$$\delta'_i = \mathbf{T}'_i - \mathbf{Q}_i = \sum_{j=0}^{n-i} B^{n-i}_j(\mu')\mathbf{R}_{i+j} - \mathbf{Q}_i \quad i = 0, 1, \ldots, n \tag{42}$$

Since we wish to obtain μ' such that optimization criterion O be minimized, we have

$$\frac{dO(\epsilon'_i, \delta'_i)}{d\mu'} = 0 \tag{43}$$

where $O(\epsilon'_i, \delta'_i)$ is the appropriate optimization criterion. Substituting Eqs. (41) and (42) into Eq. (43) will produce a non-linear equation of μ'. Practical algorithms can be derived to calculate μ' by using Newton's method.

Remark 3. *Merging methods discussed in this paper can be further extended for cases of multiple Bézier curves. Given a sequence of consecutive Bézier curves, our objective now is to obtain a new sequence of consecutive Bézier curves that best approximates the original Bézier curves set. We intend to minimize the number of curves in the new sequence with the condition that the merging error stays within a required error range* **E**. *It can proceed as follows. First we group the initial sequence into adjacent pairs, leaving the single Bézier curve at the end if the number of curves is odd. Then for each pair, we apply optimization method with endpoints matching and examine the merging error. If the error exceeds a specified error bound ϵ, the pair will not be merged and becomes two curves in the result sequence. Otherwise the pair will be merged and all merged curves from these pairs form a new sequence of Bézier curves (not necessarily consecutive) for the next round of grouping. Repeat the process until no two adjacent Bézier curves can be found in the sequence, and we finally obtain the result sequence of Bézier curves. Note that error bound of each step of merging ϵ must be carefully specified such that accumulated merging error of the process meets the requirement* **E**.

Acknowledgements

This work was supported by the Natural Science Foundation of China (project number 69902004). We thank Prof. Tong-Guang Jin and Prof. Guo-Zhao Wang for encouraging this work, and Dr Wen-Shen Xu for his useful suggestions. We especially thank the reviewers for their helpful comments.

References

[1] Dannenberg L, Nowacki H. Approximate conversion of surface representations with polynomial bases. Computer-Aided Geometric Design 1985;2:123–31.

[2] Hoschek J. Approximate conversion of spline curves. Computer-Aided Geometric Design 1987;4:59–66.

[3] Watkins MA, Worsey AJ. Degree reduction of Bézier curves. Computer-Aided Design 1988;20:398–405.

[4] Lachance MA. Chebyshev economization for parametric surfaces. Computer-Aided Geometric Design 1988;5:192–208.

[5] Eck M. Degree reduction of Bézier curves. Computer-Aided Geometric Design 1993;10:237–51.

[6] Eck M. Least square degree reduction of Bézier curves. Computer-Aided Design 1995;27:845–51.

[7] Bogacki P, Weinstein SE, Xu Y. Degree reduction of Bézier curves by uniform approximation with endpoint interpolation. Computer-Aided Design 1995;27:651–61.

[8] Hu Shi-Min, Jin Tong-Guang. Approximate degree reduction of Bézier curves. Proceedings of Symposium on Computational Geometry, Held in Hangzhou, China, 1992. p. 110–26 (also in: Tsinghua Science and Technology 1998;3:997–1000).

[9] Hu Shi-Min, Wang Guo-Zhao, Jin Tong-Guang. Properties of two types of generalized Ball curves. Computer-Aided Design 1996;28:218–22.

[10] Piegl L. Algorithm for degree reduction of B-spline curves. Computer-Aided Design 1995;27:101–10.

[11] Hu Shi-Min, Zheng Guo-Qin, Sun Jia-Guang. Approximate degree reduction of rectangular Bezier surfaces. Chinese Journal of Software Research 1997;4:353–61.

[12] Hu Shi-Min, Zuo Zheng, Sun Jia-Guang. Approximate degree reduction of triangular Bezier surfaces. Tsinghua Science and Technology 1998;3:1001–4.

[13] Zhihao Cao, Zhang Yude, et al. Matrix computation and equation solving. China: Advanced Education Press, 1984.

Shi-Min Hu is an associate professor in the Department of Computer Science and Technology at Tsinghua University. He received the BS in mathematics from the 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 postdoctoral research at Tsinghua University in 1998. His current research interests are in computer-aided geometric design, computer graphics and content-based images indexing.

Ruo-Feng Tong is an associate professor in the Department of Computer Science at Zhejiang University. He received the BS in Applied Mathematics from Zhejiang University of China in 1991, and the PhD in CAGD and Computer Graphics from the Zhejiang university of China in 1996. His current research interests are in computer-aided geometric design and computer graphics.

Tao Ju is an Undergraduate at Tsinghua University for his degree both in Computer Science and English Language. His current research interests are in computer-aided geometric design.

Jia-Guang Sun is a professor in the Department of Computer Science and Technology at Tsinghua University. He is also the Director of National CAD Engineering Center at Tsinghua University and Academician of the Chinese Academy of Engineering. He received the BS in Computer Science from Tsinghua University in 1972. His current research interests are in computer-aided geometric design, computer graphics and Product Data Management.