Effective $C^1 G^2$-merging of Two Bézier Curves by Matrix Computation

Lizheng Lu

Department of Information and Computing Science, Zhejiang Gongshang University,
luzl99@yahoo.com.cn

Abstract

The merging of polynomial curves is frequently required when modeling the complex shape in geometric design and related applications. In this paper, we present an effective method for the $C^1 G^2$-merging of two Bézier curves by using matrix computation. By minimizing the distance function defined in terms of control points, the optimal merged curve is obtained and expressed in the matrix form. Due to the two variables available from the conditions of $C^1 G^2$-continuity at the endpoints, the merged curve in this way is clearly better than that obtained by the $C^2$-merging. Numerical examples are provided to demonstrate the effectiveness of the proposed method.

Keywords: Bézier Curve, Merging, $C^1 G^2$-continuity, Matrix computation

1. Introduction

Polynomial curves in the Bernstein–Bézier representation are widely used for free-form modeling in Computer Aided Geometric Design (CAGD) and CAD [1,2]. Since every CAD system may have its own degree limit, it is frequently required to take exchange, compression, transfer and comparison between different CAD systems. While a given curve of a certain degree can be equivalently expressed as a higher degree curve, it is not easy and usually impossible to express it as a lower degree curve. So, polynomial approximation is proposed to approximate the given curve by another curve with a lower degree, which is called degree reduction and has been extensively studied, see e.g. [3-8] and the references therein.

A common task in shape design is to model complex curves. If a complex curve is constructed and modified by a single Bézier curve, it will usually turn out to be tedious and involved. So, such management limits the use and scope of CAD systems in geometric design and other interactive applications. This problem can be solved by using the merging technique. That is, many segments of curves are firstly constructed so as to represent the complex shape, and then they will be merged into a single segment by using approximation methods. Hu et al. [9] proposed to use the constrained optimization for the merging of a pair of Bézier curves. The key idea is first to derive conditions for the precise merging of Bézier curves, and then to obtain the optimization solution by moving all the control points. In contrast, Cheng and Wang [10] considered the approximate merging of multiple adjacent Bézier curves having different degrees. The solution is solved from the conditions of the unified matrix representation for the precise merging.

Inspired by the degree reduction work by using the constraint of geometric continuity [11,12], in this paper we investigate the $C^1 G^2$-merging problem, where the merged curve will keep $C^1$-continuity and $G^2$-continuity at the endpoints. Compared to the common parametric continuity, geometric continuity has the ability to provide some more variables so as to further optimize the solution [13]. Therefore, in the wide field of geometric modeling and computer graphics, many problems have been perfectly solved by using the conditions of geometric continuity. When it comes to the merging problem, the main advantage is that the approximation result can be further optimized by the additional variables provided by geometric continuity. So we can obtain the optimal merged curve with a smaller approximation error. This is why the result obtained by the $C^1 G^2$-merging is better than that obtained by the $C^2$-merging.

This paper is organized as follows. Definitions and preliminaries for Bézier curves are given in Section 2. In Section 3, we then describe the $C^1 G^2$-merging problem and define the distance function for the problem in terms of control points. In Section 4, we present an algorithm to effectively solve the $C^1 G^2$-merging problem by matrix computation. We make use of matrix representations to derive the final solution by minimizing the distance function. Numerical experiments are given in Section 5. Finally, we conclude the paper in Section 6.
2. The Bernstein-Bézier representation

A Bézier curve of degree \( n \) is defined by (see [1, 2])

\[
P(t) = \sum_{i=0}^{n} B_i^n(t) p_i, \quad 0 \leq t \leq 1,
\]

where \( B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i \) are the Bernstein polynomials, and \( p_i \) are called control points. The Bernstein-Bézier representation of polynomial curves has many good properties like the de Casteljau algorithm and elegant geometric interpretations, see the classical textbook [2] for a thorough study. For the convenience of computation and discussion, the Bézier curve is usually rewritten in the matrix form, 

\[
P(t) = B_n P_n, \text{ where}
\]

\[
B_n = (B_0^n(t), \ldots, B_n^n(t)) \quad \text{and} \quad P_n = (p_0, \ldots, p_n)^T.
\]

In practice, various Bézier curves may have different degrees. So, it is often required to unify the degrees of curves or to improve the flexibility in geometric design by adding other control points. A common way to achieve this goal is called degree elevation, which raises the degree of a curve without changing its shape. For a Bézier curve of degree \( m \) with control points \( P_m \), we can increase its degree to a higher degree \( n (> m) \) and represent it by \( B_n T_{n,m} P_m \), where the degree elevation operator \( T_{n,m} \) is an \( (n+1) \times (m+1) \) matrix with the elements given by (see [4])

\[
T_{n,m}(i, j) = \binom{m}{j} \binom{n-m}{i-j}, \quad i = 0, 1, \ldots, n, \quad j = 0, 1, \ldots, m.
\]

3. The \( C^1G^2 \)-merging problem

Let \( P(t) \) and \( Q(t) \) be two Bézier curves of degree \( n_1 \) and degree \( n_2 \),

\[
P(t) = \sum_{i=0}^{n_1} B_i^{n_1}(t) p_i = B_{n_1} P_{n_1} \quad \text{and} \quad Q(t) = \sum_{i=0}^{n_2} B_i^{n_2}(t) q_i = B_{n_2} Q_{n_2},
\]

where \( P_{n_1} = (p_0, \ldots, p_{n_1})^T \) and \( Q_{n_2} = (q_0, \ldots, q_{n_2})^T \) are their control points respectively. The \( C^1G^2 \)-merging problem is a process that amounts to find a Bézier curve of degree \( n \) with control points \( R_n = (r_0, \ldots, r_n)^T \),

\[
R(t) = \sum_{i=0}^{n} B_i^n(t) r_i = B_n R_n, \quad 0 \leq t \leq 1.
\]
such that a suitable distance function is minimized. Usually, we assume that \( n > n_1 \) and \( n > n_2 \), so as to generate satisfactory merging results. See Figure 1 for an example, where two polynomial curves (displayed in dashed and dotted lines) are merged to a single curve (displayed in solid lines).

We consider the distance function defined in terms of their control points,

\[
\varepsilon = \left\| \mathbf{S}_1 \mathbf{R} - T_{n,n_1} \mathbf{P} \right\|^2 + \left\| \mathbf{S}_2 \mathbf{R} - T_{n,n_2} \mathbf{Q} \right\|^2,
\]

where

\[
\mathbf{S}_1 = \begin{bmatrix}
1 & 0 & \cdots & 0 & 0 \\
B_0^0(\lambda) & B_1^0(\lambda) & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
B_0^{n-1}(\lambda) & B_1^{n-1}(\lambda) & \cdots & B_{n-1}^{n-1}(\lambda) & 0 \\
B_0^n(\lambda) & B_1^n(\lambda) & \cdots & B_{n-1}^n(\lambda) & B_n^n(\lambda)
\end{bmatrix}
\]

and

\[
\mathbf{S}_2 = \begin{bmatrix}
B_0^n(\lambda) & B_1^n(\lambda) & \cdots & B_{n-1}^n(\lambda) & B_n^n(\lambda) \\
0 & B_0^{n-1}(\lambda) & \cdots & B_{n-2}^{n-1}(\lambda) & B_{n-1}^{n-1}(\lambda) \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & B_0^1(\lambda) & B_1^1(\lambda) \\
0 & 0 & \cdots & 0 & 1
\end{bmatrix}
\]

are called subdivision matrices and are obtained by subdividing the target curve \( \mathbf{R}(t) \) at a user-prescribed parameter \( \lambda \in (0,1) \) via the de Casteljau algorithm, see [2]. In (1), the degree elevation operator are used to raise the degree \( n_1 \) of \( \mathbf{P}(t) \) and the degree \( n_2 \) of \( \mathbf{Q}(t) \) to the same degree \( n \). Therefore, control points can then be pairwise compared since all the related curves will have the same number of control points. By minimizing the distance function, the merged curve will be forced to approach the two given curves. So, we can determine the optimal value for all the control points \( \mathbf{R}_n \) and obtain the target curve sequently.

\( C^1G^2 \)-merging means we must satisfy the required continuity conditions at the endpoints. More precisely, the target curve \( \mathbf{R}(t) \) and the two given curves share \( C^1 \)-continuity and \( G^2 \)-continuity at the two endpoints of \( \mathbf{R}(t) \). According to the endpoint interpolation and derivatives of Bézier curves, the condition of \( C^1 \)-continuity will be satisfied if

\[
\mathbf{r}_0 = \mathbf{p}_0, \quad \mathbf{r}_n = \mathbf{q}_{n_2}
\]

and

\[
\mathbf{r}_1 = \mathbf{p}_0 + \frac{n_1}{n} (\mathbf{p}_1 - \mathbf{p}_0),
\]

\[
\mathbf{r}_{n-1} = \mathbf{q}_{n_1} + \frac{n_2}{n} (\mathbf{q}_{n_1-1} - \mathbf{q}_{n_2}).
\]

By using the conclusion in [11], the condition of \( G^2 \)-continuity will be satisfied if

\[
\mathbf{r}_2 = 2\mathbf{r}_1 - \mathbf{r}_0 + \frac{n_1(n_1 - 1)}{n(n - 1)} (\mathbf{p}_2 - 2\mathbf{p}_1 + \mathbf{p}_0) + \frac{n_1}{n(n - 1)} \eta_0 (\mathbf{p}_1 - \mathbf{p}_0)
\]

and

\[
\mathbf{r}_{n-2} = 2\mathbf{r}_{n-1} - \mathbf{r}_0 + \frac{n_2(n_2 - 1)}{n(n - 1)} (\mathbf{q}_{n_2} - 2\mathbf{q}_{n_2-1} + \mathbf{q}_{n_2+2}) + \frac{n_2}{n(n - 1)} \eta_0 (\mathbf{q}_{n_2} - \mathbf{q}_{n_2-1}).
\]

In order to obtain (5) and (6), \( \mathbf{r}_1 \) and \( \mathbf{r}_{n-1} \) must be firstly calculated by (3) and (4).
Note that the condition of $G^2$-continuity provides two degrees of freedom for geometric design, which are $\eta_0$ and $\eta_1$. Once these variables are assigned to different values, we can obtain different curves for the merging problem. So, we can further optimize the result and obtain the optimal curve by using these variables. At the meantime, the variables can also be prescribed by special values according to various applications. Then, their values are thus fixed, and the other control points will be determined by minimizing the distance function. This advantage is really required in geometric design, since it provides an interactive tool.

4. Algorithm

The $C^1G^2$-merging problem can be solved through two stages. In the first stage, we let the degree $n$ Bézier curve $\mathbf{R}(t)$ interpolate the first three and the last three control points determined in (2)–(6). That is, it can be represented by

$$
\mathbf{R}(t) = B^n_0(t)\mathbf{r}_0 + B^n_1(t)\mathbf{r}_1 + B^n_2(t)\mathbf{r}_2 + \sum_{i=3}^{n-3} B^n_i(t)\mathbf{r}_i + B^n_{n-2}(t)\mathbf{r}_{n-2} + B^n_{n-1}(t)\mathbf{r}_{n-1} + B^n_n(t)\mathbf{r}_n,
$$

where $\mathbf{r}_2$ and $\mathbf{r}_{n-2}$ contain the unknown variables $\eta_0$ and $\eta_1$ respectively (cf. (5) and (6)).

We assume that the unknown variables $\eta_0$ and $\eta_1$ are temporarily fixed, then solve the inner control points $\mathbf{r}_i, i = 3, \ldots, n-3$ by minimizing the distance function defined in (1). In order to distinguish the inner control points from the other six ones, we permute some control points in $\mathbf{R}_n$ and let

$$
\mathbf{R}^c_n = (\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_{n-2}, \mathbf{r}_{n-1}, \mathbf{r}_n)^T
$$

and

$$
\mathbf{R}^f_n = (\mathbf{r}_3, \ldots, \mathbf{r}_{n-3})^T
$$

denote the constrained control points and the unconstrained control points respectively. Then, we can easily derive that

$$
S_1^c \mathbf{R}_n = S_1^c \mathbf{R}^c_n + S_1^f \mathbf{R}^f_n
$$

and

$$
S_2^c \mathbf{R}_n = S_2^c \mathbf{R}^c_n + S_2^f \mathbf{R}^f_n,
$$

where $S_1^c = S_1(0, \ldots, n; 0, 1, 2, n-2, n-1, n)$ and $S_1^f = S_1(0, \ldots, n; 3, \ldots, n-3)$ denote submatrices of the matrix $S_1$ formed by selecting the certain columns.

Obviously, the distance function in (1) can be rewritten as

$$
\varepsilon = \left\| S_1^c \mathbf{R}^c_n + S_1^f \mathbf{R}^f_n - T_{n,\eta_1} \mathbf{P}_{n,\eta_1} \right\| + \left\| S_2^c \mathbf{R}^c_n + S_2^f \mathbf{R}^f_n - T_{n,\eta_2} \mathbf{Q}_{n,\eta_2} \right\|^2.
$$

(7)

Taking the partial derivatives of (7) with respect to $\mathbf{r}_i (i = 3, \ldots, n-3)$ and setting the derivatives equal to zero lead to

$$
M \mathbf{R}^f_n = (S_1^f)^T (T_{n,\eta_1} \mathbf{P}_{n,\eta_1} - S_1^c \mathbf{R}^c_n) + (S_2^f)^T (T_{n,\eta_2} \mathbf{Q}_{n,\eta_2} - S_2^c \mathbf{R}^c_n),
$$

where

$$
M = (S_1^f)^T S_1^f + (S_2^f)^T S_2^f.
$$

It is easy to derive that the matrix $M$ is nonsingular. Therefore, $\varepsilon$ is minimized by choosing

$$
\mathbf{R}^f_n = M^{-1} \left( (S_1^f)^T (T_{n,\eta_1} \mathbf{P}_{n,\eta_1} - S_1^c \mathbf{R}^c_n) + (S_2^f)^T (T_{n,\eta_2} \mathbf{Q}_{n,\eta_2} - S_2^c \mathbf{R}^c_n) \right).
$$

(8)

The second stage is then to determine the two unknown variables. By substituting all the control points $\mathbf{r}_i$ expressed by (2)–(6) and (8) into (7), we change the minimization of (7) to a quadratic optimization with two unknowns, which can be easily solved [14-16].

After replacing all the variables $\eta_0$ and $\eta_1$ in $\mathbf{R}_n$ with the values solved in the second stage, we obtain the merged curve $\mathbf{R}(t)$ which preserves $C^1$-continuity and $G^2$-continuity at the endpoints. From the above discussion and computation, we can finally summarize the algorithm for the $C^1G^2$-merging problem as follows.
Algorithm 1.
Input: \( P_n, Q_n, n_1, n_2, n \).
Output: \( R_n, \varepsilon \).

Step 1. Express \( r_i, i = 0, \ldots, n \) by (2)–(6) and (8) in terms of \( \eta_0 \) and \( \eta_1 \).

Step 2. Solve \( \eta_0 \) and \( \eta_1 \) by minimizing the quadratic function (7).

Step 3. Compute \( r_i, i = 0, \ldots, n \) by (2)–(6) and (8) and the distance function \( \varepsilon \) by (1).

5. Numerical experiments

In this section, we will show some examples for the \( C^1G^2 \)-merging. The merged curve, of course, relies on the value of the subdivision parameter \( \lambda \), which lies in the domain \((0, 1)\). It must be firstly prescribed by the user. And then the user can adjust its value so as to obtain better merging results in various examples.

In the first example, we consider two \( C^1 \)-joined quartic Bézier curves, which are defined by the control points \((0, 0), (0.25, 0.5), (0.5, 0), (0.75, -0.25), (1, 0) \) and \((1, 0), (1.25, 0.25), (1.5, 0), (1.75, -0.5), (2, 0)\), respectively. Figure 2 shows the merging results obtained by using Algorithm 1. We take \( \lambda = 0.5 \), since the two given curves are symmetric with respect to the joint point. The \( C^1G^2 \)-merging (displayed in red dashed lines) is better than the \( C^2 \)-merging (displayed in blue dashed lines). Therefore, better approximation can be obtained if we make the endpoints preserve \( C^2 \)-continuity, instead of \( C^1 \)-continuity. The obvious advantage of the proposed method is that it is able to provide two degrees of freedom in geometric design to obtain better results by adjusting their values according to various applications.

In the second example, we consider two Bézier curves having different degrees, whose control points are \((1.9, 2.4), (2.7, 2.9), (1.1, 6), (0, 0), (0.1, 3.1), (0.1, 0.4), (0.4, 0.4) \) and \((0.4, 0.4), (1.5, 0.3), (2.6, 1.4), (3.2, 2.9)\), respectively. Clearly, these two curves are connected with only \( C^0 \)-continuity. In Figures 3 and 4, we illustrate the merging results with the endpoints preserving \( C^1G^2 \)-continuity and \( C^2 \)-continuity, respectively. Also, we take \( \lambda = 0.5 \) for the sake of simplicity. From the results, we can obtain two observations. First, for every degree \( n \), the \( C^1G^2 \)-merging is better than the \( C^2 \)-merging. And second, the merged curves approach the two given curves rapidly with the increase of the degree \( n \). In Table 1, we list the distance of the merging result for every case.

<table>
<thead>
<tr>
<th>( n )</th>
<th>( C^1G^2 )-merging</th>
<th>( C^2 )-merging</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>3.6421</td>
<td>4.1376</td>
</tr>
<tr>
<td>10</td>
<td>1.5960</td>
<td>2.2252</td>
</tr>
<tr>
<td>12</td>
<td>0.6766</td>
<td>0.9806</td>
</tr>
<tr>
<td>14</td>
<td>0.4064</td>
<td>0.6335</td>
</tr>
<tr>
<td>16</td>
<td>0.2581</td>
<td>0.4360</td>
</tr>
</tbody>
</table>

Figure 2. Merging of two quartic Bézier curves by a single Bézier curve of degree \( n \). The merged curves with \( C^1G^2 \)-continuity and \( C^2 \)-continuity are displayed in red and blue dashed lines respectively. (a) \( n = 7 \); (b) \( n = 9 \).
Figure 3. Merging of two Bézier curves having different degrees by a single Bézier curve of degree $n$. The merged curves with $C^1G^2$-continuity are displayed in red dashed lines. (a) The two given curves; (b) $n = 8$; (c) $n = 10$; (d) $n = 12$; (e) $n = 14$; (f) $n = 16$

Figure 4. Merging of two Bézier curves having different degrees by a single Bézier curve of degree $n$. The merged curves with $C^2$-continuity are displayed in blue dashed lines. (a) The two given curves; (b) $n = 8$; (c) $n = 10$; (d) $n = 12$; (e) $n = 14$; (f) $n = 16$
5. Conclusion

In this paper, we have proposed a simple and effective method for the $C^1G^2$-merging of two Bézier curves by using matrix computation. The main idea is to minimize the distance function defined in terms of control points and the two variables available from the conditions of $C^2G^2$-continuity at the endpoints. Compared to the common $C^2$-merging, the proposed $C^1G^2$-merging has better merging results. Experiment results show the effectiveness of the proposed method.

6. Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61272307 and 11226329) and the Zhejiang Provincial Natural Science Foundation of China (Grant No. Y1110034).

7. References