Short communication

Approximate merging of multiple Bézier segments

Min Cheng a,b, Guojin Wang a,*

a Institute of Computer Graphics and Image Processing, Zhejiang University, Hangzhou 310027, China
b College of Science, Zhejiang University of Technology, Hangzhou 310032, China

Received 30 November 2007; received in revised form 15 January 2008; accepted 15 January 2008

Abstract

This paper deals with the approximate merging problem of multiple adjacent Bézier curves with different degrees by a single Bézier curve, which is a frequently seen problem in modeling. The unified matrix representation for precise merging is presented and the approximate merging curve is further derived based on matrix operation. Continuity at the endpoints of curves is also discussed in the merging process. Examples show that the method in this paper achieves satisfying merging results.

© 2008 National Natural Science Foundation of China and Chinese Academy of Sciences. Published by Elsevier Limited and Science in China Press. All rights reserved.

Keywords: Computer-aided geometric design; Bézier curve; Approximate merging; Matrix representation

1. Introduction

As different modeling systems for designs and machining in different regions are coming out ceaselessly, and more multinational corporations are being established, internet has become more and more popularized. Thus, data exchanging, data share and data integration among geometric information of various computer-aided design/manufacture systems are getting increasingly frequent [1]. The general aim when transferring geometric information from one system to the other is to ensure a high degree of accuracy, the least possible loss of information and a small amount of geometric data for communication.

For reducing the amount of communicating data, approximate conversion is the usually adoptive technique. The problem of approximate conversion has been stated in the work of Hoschek [2], who mentioned that the approximate conversion includes two basic operations: one is degree reduction, i.e. finding a parametric curve of degree \( n \) that approximates the given curve of degree \( m > n \); the other is merging, which deals with the problem of merging as many curve segments as possible to get one curve segment.

Degree reduction has been extensively investigated by many researchers [3–9]. As for merging, only a few studies have been done. Merging two adjacent Bézier curves of same degree by a single Bézier curve is discussed in Ref. [10] 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. This idea has also been introduced to solve the problem of B-spline curve merging [11]. To satisfy the more needs of approximate conversion in CAD/CAM systems, it is necessary to develop merging techniques for multiple Bézier segments whose degree may be different. This paper is just a report of the research results to address this issue.

The basic idea of this paper is to give a unified matrix representation for precise merging, and then by introducing the Moore–Penrose generalized inverse matrix theory, the approximate merging curve can be achieved based on matrix operation. Continuity at the endpoints of curves is discussed in the merging process. Examples of approximate merging of multiple Bézier segments are also given.
2. Unified matrix representation for precise merging

2.1. Problem statement

Here, we give a description of more general problem of Bézier curve merging as follows. Given \( k \) adjacent Bézier curves \( P_j(t), j = 1, 2, \ldots, k \) with degrees \( m_1, m_2, \ldots, m_k \) and control points \( P_0^{(j)}, P_1^{(j)}, \ldots, P_{m_j}^{(j)}, j = 1, 2, \ldots, k \), respectively, the curves can be represented as

\[
P_j(t) = \sum_{i=0}^{m_j} B_{m_j}^{(j)}(t) \mathbf{P}_i^{(j)}, \quad 0 \leq t \leq 1, \quad j = 1, 2, \ldots, k
\]  

(1)

where \( B_{m_j}^{(j)}(t) \), \( i = 0, 1, \ldots, m_j \) is the Bernstein basis of degree \( m_j \). The above-mentioned \( k \) adjacent Bézier curves \( P_j(t), j = 1, 2, \ldots, k \) can be restricted to the interval \([0, 1]\) via parameter transformation as follows:

\[
\tilde{P}(t) = \sum_{j=0}^{m} P_j^{(1)}B_{m}^{(1)}( \frac{t-t_j}{t_j-t_{j-1}}), \quad t_{j-1} \leq t \leq t_j
\]

(2)

where the discrete parameters \( \{t_1, t_2, \ldots, t_{k-1}\} \) can be chosen equidistant or taking the chord lengths into consideration.

Merging of \( P_j(t), j = 1, 2, \ldots, k \) amounts to the problem of finding a degree \( n \) Bézier curve \( Q(t) = \sum_{i=0}^{n} B_{n}^{i}(t) Q_i, \quad 0 \leq t \leq 1 \) with control points \( Q_0, Q_1, \ldots, Q_{n} \), such that a suitable distance between \( Q(t) \) and \( 
\tilde{P}(t) \) is minimized on the interval \([0, 1]\).

If degree \( n \) of the merging curve is much smaller than the given adjacent Bézier curves, the merging error will be usually too large to be accepted in practice. Thus in this paper, we suppose \( n \geq \max\{m_1, m_2, \ldots, m_k\} \). If the customers do require \( n < \max\{m_1, m_2, \ldots, m_k\} \), the method in this paper can be modified by using a degree elevation to satisfy the demand.

2.2. Discrete property for Bézier curve

A Bézier curve is usually defined on the interval \([0, 1]\), but it can also be defined on any interval \([a, b]\) \( 0 \leq a, b \leq 1 \). The part of the curve that corresponds to \([a, b]\) can also be defined by a Bézier polygon, and its control points can be derived from the original curve.

For a degree \( n \) Bézier curve \( Q(t) = \sum_{i=0}^{n} B_{n}^{i}(t) Q_i, \quad 0 \leq t \leq 1 \), its sub-curve on \([a, b]\) \( 0 \leq a, b \leq 1 \) can also be represented in Bézier form of the same degree as follows:

\[
Q(t) = \sum_{j=0}^{n} B_{n}^{i}(t) (Q_j - aQ_{j-1} + bQ_{j+1}), \quad 0 \leq t \leq 1
\]

where

\[
\begin{array}{c}
Q_0 \\
Q_1 \\
\vdots \\
Q_{n-1} \\
Q_n
\end{array}
\]

(3)

\[
\begin{array}{c}
B_{n}^{0}(\frac{t}{b-a}) \\
B_{n}^{1}(\frac{t}{b-a}) \\
\vdots \\
B_{n}^{n}(\frac{t}{b-a})
\end{array}
\]

is an upper triangular matrix and

\[
\begin{array}{c}
0 \\
B_{n}^{1}(\frac{t}{b-a}) \\
\vdots \\
B_{n}^{n}(\frac{t}{b-a})
\end{array}
\]

is a lower triangular matrix.

Thus, we can see from the above conclusion that the new control points \( \{Q_0, Q_1, \ldots, Q_{n-1}, Q_n\} \) of the sub-curve on \([a, b]\) can be calculated by multiplying the original control points \( \{Q_0, Q_1, \ldots, Q_{n-1}, Q_n\} \) with two conversion matrix, i.e.

\[
\begin{bmatrix}
Q_0 \\
Q_1 \\
\vdots \\
Q_{n-1} \\
Q_n
\end{bmatrix}
= C_{\text{upper}}^{(a/b)} C_{\text{lower}}^{(b)}
\begin{bmatrix}
Q_0 \\
Q_1 \\
\vdots \\
Q_{n-1} \\
Q_n
\end{bmatrix}
\]

(4)

This subdivision process can be illustrated by Fig. 1.

2.3. Unified matrix representation for precise merging

If the given \( k \) adjacent Bézier curves \( P_j(t), j = 1, 2, \ldots, k \) mentioned in (1) can be merged precisely, that is to say
there exists a degree \( n \) Bézier curve \( Q(t) = \sum_{i=0}^{n} B^i_n(t)Q_i \), \( 0 \leq t \leq 1 \) such that \( Q(t) \equiv P(t) \), where \( P(t) \) is defined in (2). In other words, the \( k \) sub-curves of \( Q(t) \) on \([0,t_1],[t_1,t_2],...,[t_{k-2},t_{k-1}], [t_{k-1},1]\) happen to match precisely with the given \( k \) adjacent Bézier curves \( P(t) \), \( j = 1,2,...,k \) defined in (1). Here, we might as well suppose that the discrete parameters \([t_1,t_2,...,t_{k-1}]\) have already been chosen. Now, we give the conditions for precise merging for each subsection as follows.

2.3.1. The first subsection on \([0,t_1]\)

Applying the discrete property for Bézier curve, which is mentioned in Section 2.2, we get

\[
\sum_{i=0}^{n} B^i_n(t)Q_i = \sum_{i=0}^{n} B^i_n\left(\frac{t}{t_1}\right)Q_i, \quad 0 \leq t \leq t_1
\]  

(5)

where \( Q_i = \sum_{j=0}^{i} B_j^i(t_1)Q_j \).

Using matrix representation, the above conversion can be expressed as

\[
[B_0^0(t), B_1^0(t), ..., B_n^0(t)] \times [Q_0, Q_1, ..., Q_n] = [Q_0, Q_1, ..., Q_n]
\]

where the matrix \( C_n^{\text{lower}}(t_1) \) is defined in (3).

The precise merging acquires the first subsection of \( Q(t) \) on \([0,t_1]\), i.e. \( Q(t) = \sum_{i=0}^{n} B^i_n(t)Q_i \), \( 0 \leq t \leq t_1 \) matches \( P_i(t) = \sum_{i=0}^{m} P^i_n(t)P_i^{(k)} \), \( 0 \leq t \leq 1 \). Using degree elevation, we can get the precise merging conditions for the first sub-curve.

Denote

\[
A_n = \begin{bmatrix}
1 & 0 & 0 & \ldots & 0 & 0 \\
\frac{1}{n} & \frac{n-1}{n} & 0 & \ldots & 0 & 0 \\
0 & \frac{2}{n} & \frac{n-2}{n} & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & \frac{n-1}{n} & \frac{1}{n} \\
0 & 0 & 0 & \ldots & 0 & 1
\end{bmatrix}
\]

as the \((n+1) \times n\) degree elevation matrix. By multiplying the control points vectors with the degree elevation matrices, we can get the degree elevated curve of \( P_i(t) \) as \( A_nA_{n-1} \cdots A_{m_1+2}A_{m_1+1}P_i^{(k)}(t) \), which is a degree \( n \) Bézier curve now. As it matches \( Q(t) = \sum_{i=0}^{n} B^i_n(t)Q_i \), \( 0 \leq t \leq t_1 \), we get the precise merging conditions for the first sub-curve as

\[
A_nA_{n-1} \cdots A_{m_1+2}A_{m_1+1} = C_n^{\text{lower}}(t_1)
\]

(7)

2.3.2. The last subsection on \([t_{k-1},1]\)

For the last subsection on \([t_{k-1},1]\), we have

\[
\sum_{i=0}^{n} B^i_n(t)Q_i = \sum_{i=0}^{n} B^i_n\left(\frac{t-t_{k-1}}{1-t_{k-1}}\right)Q_i^{n-i}, \quad t_{k-1} \leq t \leq 1
\]

(8)

where \( Q_i^{n-i} = \sum_{j=0}^{n-i} B_j^{n-i}(t_{k-1})Q_j \).

Using matrix representation, the above conversion can be expressed as

\[
[B_0^0(t), B_1^0(t), ..., B_n^0(t)] \times [Q_0, Q_1, ..., Q_n] = [Q_0, Q_1, ..., Q_n]
\]

where the matrix \( C_n^{\text{upper}}(t_{k-1}) \) is defined in (3).

The precise merging acquires the last subsection of \( Q(t) \) on \([t_{k-1},1]\), i.e. \( Q(t) = \sum_{i=0}^{n} B^i_n(t)Q_i \), \( t_{k-1} \leq t \leq 1 \) matches \( P_i(t) = \sum_{i=0}^{m} P^i_n(t)P_i^{(k)} \), \( 0 \leq t \leq 1 \). Similarly using degree elevation, we can get the precise merging conditions for the last sub-curve as

\[
A_nA_{n-1} \cdots A_{m_1+2}A_{m_1+1} = C_n^{\text{upper}}(t_{k-1})
\]

(10)

2.3.3. The \( h \)th subsection on \([t_{h-1},t_h]\)

For the \( h \)th subsection on \([t_{h-1},t_h]\), applying the discrete property for Bézier curve mentioned in Section 2.2, we have the matrix form as
\[ \sum_{i=0}^{n} B_i^a(t) Q_i = \begin{bmatrix} B_0^a(t), & B_1^a(t), & \ldots, & B_n^a(t) \end{bmatrix} \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_n \end{bmatrix} = \begin{bmatrix} B_0(t) \\ B_1(t) \\ \vdots \\ B_n(t) \end{bmatrix} \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_n \end{bmatrix} = \begin{bmatrix} B_0 \left( \frac{i - \frac{2h}{n+1}}{1 - \frac{2h}{n+1}} \right) B_1 \left( \frac{i - \frac{2h}{n+1}}{1 - \frac{2h}{n+1}} \right) \ldots \ldots B_n \left( \frac{i - \frac{2h}{n+1}}{1 - \frac{2h}{n+1}} \right) \end{bmatrix} \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_n \end{bmatrix} \times C_n^{upper} \left( \frac{t_{h-1}}{t_h} \right) C_n^{lower} \left( t_h \right) \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_n \end{bmatrix} \begin{bmatrix} C_n^{lower} \left( t_h \right) \\ \vdots \\ C_n^{lower} \left( t_{k-1} \right) \end{bmatrix} \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_n \end{bmatrix} \end{array} \right]

Thus, the precise merging condition acquires

\[ A_n A_{n-1} \ldots A_{m_1 + 2} A_{m_1 + 1} = C_n^{upper} \left( \frac{t_{h-1}}{t_h} \right) C_n^{lower} \left( t_h \right) \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_n \end{bmatrix} \begin{bmatrix} C_n^{lower} \left( t_h \right) \\ \vdots \\ C_n^{lower} \left( t_{k-1} \right) \end{bmatrix} \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_n \end{bmatrix} \]

3. Approximate merging

As the conditions for precise merging for each subsection have been given in Section 2.3, taking together, we can simultaneously get the following equations:

\[ \begin{array}{c}
A_n A_{n-1} \ldots A_{m_1 + 2} A_{m_1 + 1} = C_n^{lower} \left( t_h \right) \\
A_n A_{n-1} \ldots A_{m_2 + 2} A_{m_2 + 1} = C_n^{upper} \left( \frac{t_{h-1}}{t_h} \right) C_n^{lower} \left( t_h \right) \\
A_n A_{n-1} \ldots A_{m_k + 2} A_{m_k + 1} = C_n^{upper} \left( \frac{t_{h-1}}{t_h} \right) C_n^{lower} \left( t_h \right)
\end{array} \]

Provided that the discrete parameters \( t_1, t_2, \ldots, t_{k-1} \) have already been chosen, all the elements in (14) are known except \( Q_0, Q_1, \ldots, Q_n \). Here, we introduce the Moore–Penrose inverse theory to solve this problem [12]. The Moore–Penrose inverse has a wide application in the study of an inconsistent linear system \( Ax = b \).
Since there is no exact solution to the inconsistent linear system, least squares solution which minimizes $\|Ax - b\|_2$ is considered. With the help of matrix algebra it is not difficult to see that the solution to the normal equations $A^TAx = A^Tb$ would be a least squares solution. Let $A^+$ be the Moore–Penrose inverse of matrix $A$, then the minimal least squares solution can be obtained by $x = A^+b = (A^+A)^{-1}A^Tb$. That is to say, the control points of the merged curve $Q_0, Q_1, \ldots, Q_n$ can be calculated by the following matrix operation:

$$
[Q_0; Q_1, \ldots, Q_{n-1}; Q_n]^T = (Q_0^T Q_0)^{-1} Q_0^T Q_n
$$

Denote

$$
A = A_{k(n + 1) \times (n + 1)} = 
\begin{bmatrix}
C_{n}^{\text{lower}}(t_1) & C_{n}^{\text{upper}}(t_1) & C_{n}^{\text{lower}}(t_2) \\
\vdots & \ddots & \vdots \\
C_{n}^{\text{upper}}(t_k) & \ldots & C_{n}^{\text{lower}}(t_k)
\end{bmatrix}
$$

$$
Q = \begin{bmatrix}
Q_0 \\
Q_1 \\
\vdots \\
Q_{n-1} \\
Q_n
\end{bmatrix} = \begin{bmatrix}
P_0^{(1)} \\
P_1^{(1)} \\
\vdots \\
P_{m_k}^{(1)}
\end{bmatrix}
$$

$$
P = \begin{bmatrix}
A_{n-1} A_{n-1} \cdots A_{m_k + 2A_{m_k + 1}} [P_0^{(1)}, P_1^{(1)}, \ldots, P_{m_k}^{(1)}]^T \\
A_{n-1} A_{n-2} A_{n-2} \cdots A_{m_k + 2A_{m_k + 1}} [P_0^{(2)}, P_1^{(2)}, \ldots, P_{m_k}^{(2)}]^T \\
\vdots \\
A_{n-1} A_{n-1} \cdots A_{m_k + 2A_{m_k + 1}} [P_0^{(k)}, P_1^{(k)}, \ldots, P_{m_k}^{(k)}]^T
\end{bmatrix}
$$

(15)

$$
A = A_{k(n + 1) \times (n + 1)}
$$

(16)

Here, $A^L, A^R$ are matrices of order $(k(n + 1) - 2) \times 1$, $A^C$ is an order $(k(n + 1) - 2) \times (n - 1)$ matrix with column full rank, $Q^C$ is an order $(n - 1) \times 1$ matrix, and $P^C$ is an order $(k(n + 1) - 2) \times 1$ matrix.

By the multiplication of partitioned matrices, it is obvious that the over-determined linear equations $AQ = P$ can be deduced as

$$
A^C Q^C = P^C = A^L P_0^{(1)} - A^R P_{m_k}^{(1)}.
$$

(17)

As matrix $A^C$ is column full rank, it has the generalized inverse matrix with order $(n - 1) \times (k(n + 1) - 2)$ as

$$
(A^C)^+ = (A^C)^T A^C A^C)^{-1} A^C)^T
$$

(18)

Thus, the least squares solution of the above over-determined linear equations can be represented as

$$
Q^C = (Q_0, Q_2, \ldots, Q_{n-1})^T
$$

$$
= (A^C)^+ (P^C - A^L P_0^{(1)} - A^R P_{m_k}^{(1)})
$$

$$
= (A^C)^+ (A^L, (A^C)^+, -(A^C)^+ A^R) (P_0^{(1)}, P^C, P_{m_k}^{(1)})^T
$$

(19)

(20)

Putting the endpoints together, we get the solution of merging with endpoints continuity as

$$
Q = (Q_0, Q_1, \ldots, Q_n)^T = A^T P
$$

(21)

(22)
where
\[
A^+ = \begin{pmatrix}
1 & 0 & 0 \\
-(A^C)^+A^R & (A^C)^+ & -(A^C)^+A^R \\
0 & 0 & 1
\end{pmatrix}
\] (23)

5. Examples

Now, we give some examples to illustrate our method. As endpoints continuity is a common requirement, our presented examples all give approximation merging results with endpoints continuity.

**Example 1.** Given two adjacent cubic Bézier curves, the control points of the curves are \([-1.6, 0], [-1, 1.2], [0.3, 1.2], [0.6, 1.5] \) and \([0.6, 1.5], [1.1, 1.8], [1.8, 1.3], [1.6, 0]\). If using a degree four Bézier curve to merge the above two adjacent curves, the control points of the new merging curve are \([-2, 1], [-1, 3], [1, 2.8], [1.5, 1]\). That of the second are \([1.5, 1], [2.5, -1.5], [3.5, -2], [5, -0.5]\). And the control points of the last curve are \([5, -0.5], [6, -0.5], [8, 2], [9, 0], [11, -1]\). If using a degree six Bézier curve to merge the above three adjacent curves, the control points of the new merging curve are \([-2, 1], [-0.567651, 3.39528], (4.17486, 7.46103], [0.469161, -14.1762], [4.853, 4.02895], [7.14278, 1.71632], [11, -1]\). In Fig. 2, the original curves are rendered in solid line, with the first in black, second in blue and the last in green. The new merging curve is rendered in dotted line in red. And the error is 0.130224.

**Example 2.** Given three adjacent Bézier curves of degree 3, 3, 4, respectively. The control points of the first curve are \([-2, 1], [-1, 3], [1, 2.8], [1.5, 1]\). That of the second are \([1.5, 1], [2.5, -1.5], [3.5, -2], [5, -0.5]\). And the control points of the last curve are \([5, -0.5], [6, -0.5], [8, 2], [9, 0], [11, -1]\). If using a degree six Bézier curve to merge the above three adjacent curves, the control points of the new merging curve are \([-2, 1], [-0.567651, 3.39528], (4.17486, 7.46103], [0.469161, -14.1762], [4.853, 4.02895], [7.14278, 1.71632], [11, -1]\). In Fig. 3, the original curves are rendered in solid line, with the first in black, second in blue and the last in green. The new merging curve is rendered in dotted line in red. And the error is 0.130224.

6. Conclusion

This paper presents an algorithm for an approximate merging of multiple Bézier curves. We derive the precise merging conditions in a matrix form. The minimal least squares solution can be directly obtained. The resulting merging curve is easy to be obtained.

Acknowledgements

This work was supported by the National Basic Research Program of China (Grant No. 2004CB719400), National Natural Science Foundation of China (Grant Nos. 60673031 and 60333010) and the Scientific Research Program of the Education Office of Zhejiang Province (Grant No. 20070309).

References