G^2 Blending Ball B-Spline Curve by B-Spline

YUMING ZHAO, Beijing Normal University, China
ZHONGKE WU*, Beijing Normal University, China
XINGCE WANG, Beijing Normal University, China
XINYUE LIU, Beijing Normal University, China

Blending two Ball B-Spline Curves(BBSC) is an important tool in modeling tubular objects. In this paper, we propose a new BBSC blending method. Our method has the following three main contributions: First, we use BBSC instead of ball Bézier to model the blending part to expand the solution space and make the resultant BBSC have better fairness. Second, we consider both the skeleton line and radius of BBSC, which makes the skeleton line and radius consistent. Thirdly, we propose a two-step optimization process to solve the problem of excessive amount of parameters brought by expanding the solution space, so that our method satisfies the real-time.

CCS Concepts: • **Computing methodologies → Parametric curve and surface models**.

Additional Key Words and Phrases: G^2 continuity , Ball B-Spline Curve , curve blending , minimal strain energy

ACM Reference Format:
Yuming Zhao, Zhongke Wu, Xingce Wang, and Xinyue Liu. 2023. G^2 Blending Ball B-Spline Curve by B-Spline. *Proc. ACM Comput. Graph. Interact. Tech.* 6, 1 (May 2023), 16 pages. https://doi.org/10.1145/3585504

1 INTRODUCTION

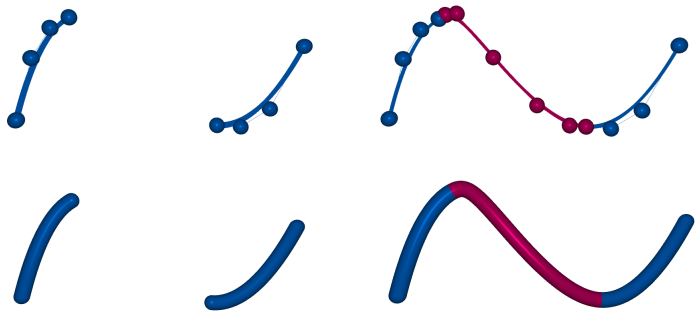
Ball B-Spline Curve (BBSC) is a method for modeling free tubular objects based on skeleton lines[Seah 2007; Seah and Wu 2005]. In BBSC, control balls are used instead of control points, and radii are added to the B-Spline curves, thus enabling a good representation of free tubular objects. BBSC is a solid representation model that can represent any point inside an object, and in addition, due to its self-contained skeleton line property, it avoids the difficult task of skeleton line extraction in the solid model processing.

Curve blending is an important task in computer-aided design(CAD). The problem is as follows: given curves C_1 and C_2 are blended into the same curve, keeping the original part of the curve unchanged and the resulting curve continuous everywhere. Fig.1 shows an example of BBSC blending task. There are two types of current curve blending methods: the first type only models the blending curve without solving for the optimal curve, and obtains the final blending curve results by providing user-adjustable parameters[Hartmann 2001; Yawen 2011]. This type of method provides the user with a high degree of freedom, but the parameter adjustment is often not intuitive, and it is difficult for the user to get the optimal blending curve results. Another type of method

*Corresponding author

Authors' addresses: Yuming Zhao, yumingzhao@mail.bnu.edu.cn, Beijing Normal University, Beijing, China; Zhongke Wu, Beijing Normal University, Beijing, China, zwu@bnu.edu.cn; Xingce Wang, Beijing Normal University, Beijing, China, wangxingce@bnu.edu.cn; Xinyue Liu, Beijing Normal University, Beijing, China, 201821210040@mail.bnu.edu.cn.

2577-6193/2023/5-ART $15.00
https://doi.org/10.1145/3585504



(a) Input ball B-Spline curves(BBSCs). (b) Result ball B-Spline curve after blending.

Fig. 1. An example of ball B-Spline curve(BBSC) blending task. Blending the input BBSCs by adding a new BBSC and maintain G^2 continuity at their junctions.

solves the optimal curve, but reduces the difficulty of solving the problem by reducing the curve degrees of freedom or by using only a single-segment Bezier curve in a limited solution space for optimal curve solving[Belkhatir et al. 2008; Liu1 et al. 2009] This type of method limits the solution space of the optimal curve, and the fairness of the obtained blending curve is often not good enough.

For the blending problem of BBSC, Jiang considers BBSC as two parts, skeleton line and radius, and blends the two parts separately[Jiang et al. 2014]. In the skeleton line blending, the method of Yongjin Liu is borrowed[Liu1 et al. 2009], and the skeleton line blending is carried out using the extension-based approach, and the energy function is optimized for minimization. In the radius blending, the strain energy of the radius function is minimized to obtain the optimal control ball radius taking value. This method has two problems: firstly, the skeleton line blending based on extending is essentially a single-segment Bezier curve for modeling the blending curve, and the solution space is limited; secondly, optimizing the skeleton line and radius separately makes the skeleton line and radius unable to maintain consistency, and the overall fairness of BBSC is poor.

Our main contributions are as follows:

1. We use ball B-Spline instead of ball Bezier as the modeling method for the blending part. Using multi-segment polynomial instead of single-segment polynomial, we obtain a larger solution space and achieve results with better fairness.

2. We consider both skeleton line and radius, the control ball is considered as a point in the 4-dimensional space, which is involved in the blending process at the same time. Compared with considering the skeleton line and radius separately, our method can be optimized in the same space to obtain the consistency of the control ball position and radius.

3. We propose a two-step optimization process to reduce the complexity of the optimization problem. Compared with the Ball Bezier curve, the modeling of the Ball B-Spline Curve as a blending part introduces more parameters. In order to avoid the optimization difficulties caused by the excessive amount of parameters, we preprocess all the control balls by the first step and express them uniformly in the form about the continuity degrees of freedom. Thus, the number of parameters in the second optimization step is controlled to be constant 4, which avoids the optimization difficulties caused by the excessive amount of parameters.

2 RELATED WORK

In this section, we review Ball B-Spline curves and the existing algorithms for belnding of B-Spline curves and BBSC.

2.1 Ball B-Spline Curve

Ball B-Spline curve (BBSC) is a skeleton line-based representation of a 3D solid model first proposed by Seah and Wu in 2005[Seah and Wu 2005]. The definition of BBSC is divided into two parts. The B-Spline curve defines the skeleton line of the solid model, and the scalar form of a B-Spline curve defines the radius of each point extending from the skeleton line. Compared with traditional Non-Uniform Rational B-Splines (NURBS), BBSC can represent curves with thickness. Therefore, BBSC has a significant advantage when representing tubular solid models. In addition, when representing the solid model, commonly used methods such as mesh model and point cloud model all need to store a large amount of data. BBSC only needs to store the coordinates and radius of control balls, as well as the knot vector. When editing the shape of the geometric model, the model expressed by using BBSC only needs to manipulate the position and radius of the control balls to complete, which is both flexible and convenient. With these excellent properties, BBSC is now widely used in realistic models of natural objects, such as 3D trees[Wu, Zhou, and Wang 2009; Wu, Zhou, Wang, et al. 2006], realistic rendering of flowering rape[L. Zhao et al. 2011], and 3D modeling of other plants [Tang et al. 2009; Zhu et al. 2008]. In the field of medical imaging, BBSC can be applied to the reconstruction and repair of cerebral vascular models[Wang, E. Liu, et al. 2016; Wang, Wu, et al. 2016; S. Zhao et al. 2010]. In 3D animation production and game development, BBSC is also useful in rapid 3D character modeling[Xu et al. 2011] and real-time animation rendering[Ao et al. 2009].

2.2 Blending of B-Spline Curve and BBSC

The problem of blending of B-Spline curve is as follows: given curves C_1 and C_2 are blended into the same curve, keeping the original part of the curve unchanged and the resulting curve continuous everywhere.

For the blending problem of B-Spline, many algorithms have been proposed. These methods can be divided into two kinds: first one only models the blending curve without performing a parametric solution of the optimal curve. The final curve is determined by providing user-adjustable parameters. Lu[Yawen 2011] adjusted the shape of the blending curves by changing the parameters lambda and thumb, ensuring that the blending results satisfy the G^2 continuity condition. Erich Hartmann[Hartmann 2001] also adjusted the shape of curves by adjusting lambda and thumb, and the G^2 continuity condition was extended to the G^n continuity condition. The other class of methods solves the optimal curve by optimizing the strain energy function. Bachir[Belkhatir et al. 2008] modeled the blending part and then optimized the solution, but only G^1 continuous is satisfied. Liu[Liu1 et al. 2009] introduced the extension method to implement the blending task and optimized it for the strain energy function of the curve.

For the blending problem of BBSC, it is more complex compared with blending problem of B-Spline curve, and there are fewer related studies. Jiang[Jiang et al. 2014] borrows the idea of Liu's method[Liu1 et al. 2009] to achieve the blending of two BBSCs . They solved the blending problem of BBSCs with the same idea as solving the extension problem of BBSCs, still extending the first BBSC to three balls by dividing the skeleton line and radius function, and then adjusting the three control balls at the end of the resultant curve according to the smooth condition with the second BBSC. Therefore, the problem of solution space limitation is not solved in any of their methods. Liu[X. Liu et al. 2020] use B-Spline as the model method to solve the curve extending task in order

to enlarge the solution space. Liu's work inspired us how to solve the the problem of solution space limitation.

3 PRELIMINARIES

Before introducing the method, we briefly introduce Ball B-Spline Curve, G^n Continuity and Matrix Representation of B-Splines.

3.1 Ball B-Spline Curve

A Ball B-Spline Curve(BBSC) is defined as

$$\langle \mathbf{B} \rangle(t) = \sum_{i=0}^{n} N_{i,k}(t) \langle \mathbf{P}_i; r_i \rangle, \quad t_{k-1} \leq t \leq t_{l-k+1} \tag{1}$$

where $N_{i,k}(t)$ is the $i-th$ B-Spline basis function of order k with knot vector $t_0, ..., t_{n+k}$, $\langle \mathbf{P}_i; r_i \rangle$ is a ball centred at P_i with radius r_i.

For a Ball B-Spline curve, it can be derived as follows:

$$\langle \mathbf{B} \rangle(t) = \sum_{i=0}^{n} N_{i,k}(t) \langle \mathbf{P}_i; r_i \rangle = \left\langle \sum_{i=0}^{n} N_{i,k}(t)\mathbf{P}_i; \sum_{i=0}^{n} N_{i,k}(t) r_i \right\rangle \tag{2}$$

According to the above equation, it can be found that as a skeleton line-based solid model, a BBSC can be viewed as two parts: a B-Spline curve as a skeleton line $\sum_{i=0}^{n} N_{i,k}(t)\mathbf{P}_i$, and a B-Spline scalar function as a radius function $\sum_{i=0}^{n} N_{i,k}(t) r_i$.

To facilitate the simultaneous consideration of the skeleton line and radius, we define the ball in terms of a 4-dimensional vector.

$$\mathbf{Q}_i = (x_i, y_i, z_i, r_i) = \langle \mathbf{P}_i; r_i \rangle \tag{3}$$

Then, a BBSC can be re-expressed as

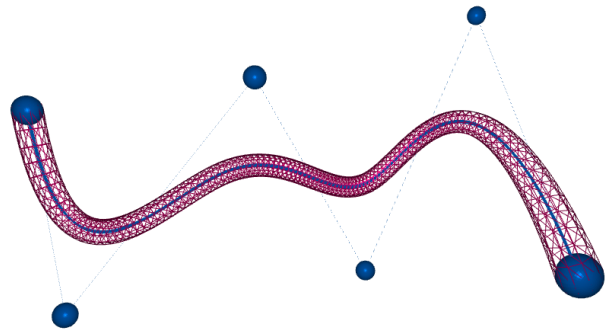
$$\langle \mathbf{B} \rangle(t) = \sum_{i=0}^{n} N_{i,k}(t) \mathbf{Q}_i \tag{4}$$

3.2 G^n Continuity

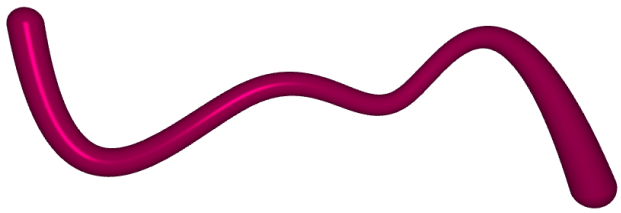
In geometry modeling, there are two types of continuity of curves: C-continuity and G-continuity. C-continuity disallows many parameterizations that generate geometrically smooth curves. G-continuity requires only that the derivatives of each order are in the same direction on both sides of a point on the curve. In the blending task, G-continuity is generally chosen as the continuity condition to give a more flexible expression of the blending curve. For two cubic B-Spline curves $\mathbf{c}_1$ and $\mathbf{c}_2$, the intersection points are $\mathbf{c}_1(1)$ and $\mathbf{c}_2(0)$, and the G-continuity condition is

$$\begin{cases} \mathbf{c}_1(0) = \mathbf{c}_2(1) \\ \mathbf{c'}_1(0) = \alpha_1 \mathbf{c'}_2(1) \\ \mathbf{c''}_1(0) = \alpha_1^2 \mathbf{c''}_2(1) + \alpha_2 \mathbf{c'}_2(1) \\ \quad \vdots \\ \mathbf{c}_1^{(n)}(0) = \alpha_1^n \mathbf{c}_2^{(n)}(1) + \cdots + \alpha_n \mathbf{c}_2^{(1)}(1) \end{cases} \tag{5}$$

where α_1 is a positive real number and $\alpha_2 ... \alpha_n$ are arbitrary real numbers.



(a) Subdivision and skeleton of BBSC.



(b) Rendering result of BBSC.

Fig. 2. Example of BBSC.

In practical applications, G^2 continuity can already satisfy the continuity requirement in general, so this paper mainly discusses the curve blending problem under the G^2 continuity condition. For $\mathbf{C}_1$ and $\mathbf{C}_2$, the bending curve is $\mathbf{B}$. Then, the expression of the G^2 continuity condition is as follows:

$$\begin{cases} \mathbf{B}(0) = \mathbf{C}_1(1) \\ \mathbf{B}'(0) = \alpha_1 \mathbf{C}'_1(1) \\ \mathbf{B}''(0) = \alpha_1^2 \mathbf{C}''_1(1) + \beta_1 \mathbf{C}'_1(1) \\ \mathbf{B}(1) = \mathbf{C}_2(0) \\ \mathbf{B}'(1) = \alpha_2 \mathbf{C}'_2(0) \\ \mathbf{B}''(1) = \alpha_2^2 \mathbf{C}''_2(0) + \beta_2 \mathbf{C}'_2(0) \end{cases} \tag{6}$$

3.3 Matrix Representation of B-Splines

The B-Spline basis function is essentially a segmented polynomial with order $k-1$. When $t \in [t_i, t_{i+1}), t_i < t_{i+1}$, there are k B-Spline basis functions with order $k-1$ that are nonzero:

$$\mathbf{N}_{i-k+1,k}(t), \mathbf{N}_{i-k+2,k}(t), ..., \mathbf{N}_{i,k}(t) \tag{7}$$

These B-Spline basis functions can be represented in matrix form as follows:

$$(\mathbf{N}_{i-k+1,k}(u), \mathbf{N}_{i-k+2,k}(u), ..., \mathbf{N}_{i,k}(u)) = (1\ u\ u^2\ ...\ u^{k-1})\mathbf{M}^k(i) \tag{8}$$

where u is a renormalization of the original parameter t, $u = (t - t_i)/(t_{i+1} - t_i), u \in [0, 1)$,

$$\mathbf{M}^k(i) = \begin{pmatrix} \mathbf{N}^k_{0,i-k+1} & \mathbf{N}^k_{0,i-k+2} & \cdots & \mathbf{N}^k_{0,i} \\ \mathbf{N}^k_{1,i-k+1} & \mathbf{N}^k_{1,i-k+2} & \cdots & \mathbf{N}^k_{1,i} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{N}^k_{k-1,i-k+1} & \mathbf{N}^k_{k-1,i-k+2} & \cdots & \mathbf{N}^k_{k-1,i} \end{pmatrix}$$

Let $\mathbf{P}$ be the control point. From the local support of the B-Spline curve, it follows that any point defined on the interval $[t_i, t_{i+1})$ can be obtained by the following equation:

$$\mathbf{B}_{i-k+1}(t) = (\mathbf{N}_{i-k+1,k}(u) \;\; \mathbf{N}_{i-k+2,k}(u) \;\; \ldots \;\; \mathbf{N}_{i,k}(u)) \cdot (\mathbf{P}_{i-k+1} \;\; \mathbf{P}_{i-k+1} \;\; \ldots \;\; \mathbf{P}_{i-k+1}) \tag{9}$$

The general form of the B-Spline matrix expression can be obtained as follows:

$$\mathbf{B}_{i-k+1}(t) = (1 \;\; u \;\; u^2 \;\; \ldots \;\; u^{k-1}) \cdot \mathbf{M}^k(i) \cdot (\mathbf{P}_{i-k+1} \;\; \mathbf{P}_{i-k+1} \;\; \ldots \;\; \mathbf{P}_{i-k+1}) \tag{10}$$

$\mathbf{M}^k(i)$ is called the coefficient matrix of the B-Spline on the interval $[t_i, t_{i+1})$ with order $k-1$. In particular, the coefficient matrix of the cubic B-Spline has the following form

$$\begin{pmatrix} \frac{(t_{i+1})^2}{(t_{i+1}-t_{i-1})(t_{i+1}-t_{i-2})} & 1-m_{00}-m_{02} & \frac{(t_i-t_{i-1})^2}{(t_{i+2}-t_{i-1})t_{i+1}-t_{i-1})} & 0 \\ -3m_{00} & 3m_{00}-m_{12} & \frac{3(t_{i+1}-t_i)(t_i-t_{i-1})}{(t_{i+2}-t_{i-1})(t_{i+1}-t_{i-1})} & 0 \\ 3m_{00} & -3m_{00}-m_{22} & \frac{3(t_i+t_i)^2}{(t_{i+2}-t_{i-1})(t_{i+1}-t_{i-1})} & 0 \\ -m_{00} & m_{00}-m_{32}-m_{33} & m_{32} & \frac{(t_{i+1}-t_i)^2}{(t_{i+3}-t_i)(t_{i+2}-t_i)} \end{pmatrix} \tag{11}$$

where m_{ij} denotes the element in row i+1 and column j+1, and $m_{32} = -\frac{1}{3}m_{22} - m_{33} - \frac{(t_{i+1}-t_i)^2}{(t_{i+2}-t_i)(t_{i+2}-t_{i-1})}$.

4 PROBLEM FORMULATION

Given two Ball B-Spline curves $\mathbf{C}_1(t)$ and $\mathbf{C}_2(t)$, our goal is to find a curve $\mathbf{B}(t)$ that allows smooth blending of $\mathbf{C}_1(t)$ and $\mathbf{C}_2(t)$. G^2continuity needs to be satisfied at the connection of $\mathbf{C}_1(t)$, $\mathbf{C}_2(t)$, and $\mathbf{B}(t)$. The strain energy function describes the fairness of $\mathbf{B}(t)$. In general, the lower the strain energy of a BBSC, the greater its fariness and the more natural its appearance. We want to find a blending curve that minimizes the strain energy while satisfying the condition of G^2continuity. Therefore, the blending problem of the B-Spline curve can be viewed as a constrained optimization problem and described as follows:

$$\begin{aligned} \min_{\mathbf{B}(t)} \quad & EG(\mathbf{B}(t)) \\ \text{s.t.} \quad & \mathbf{B}(0) = \mathbf{C}_1(1) \\ & \mathbf{B}'(0) = \alpha_1 \mathbf{C}'_1(1) \\ & \mathbf{B}''(0) = \alpha_1^2 \mathbf{C}''_1(1) + \beta_1 \mathbf{C}'_1(1) \\ & \mathbf{B}(1) = \mathbf{C}_2(0) \\ & \mathbf{B}'(1) = \alpha_2 \mathbf{C}'_2(0) \\ & \mathbf{B}''(1) = \alpha_2^2 \mathbf{C}''_2(0) + \beta_2 \mathbf{C}'_2(0) \end{aligned} \tag{12}$$

In previous work, the blending curve is usually represented by bezier with one polynomial segment. The G^2continuity at the connection is often given only a limited number of degrees of freedom. This gives a relatively small solution space for the blending curve, making it likely that the result of the solution is not the globally optimal blending curve. We use a multisegment polynomial, i.e., a B-Spline curve, as the expression of the blending curve.

5 STRAIN ENERGY FUNCTION OF BALL B-SPLINE

In this chapter, we present the definition of the strain energy function of BBSC and its discrete expression based on the control balls.

5.1 Defination of Strain Energy

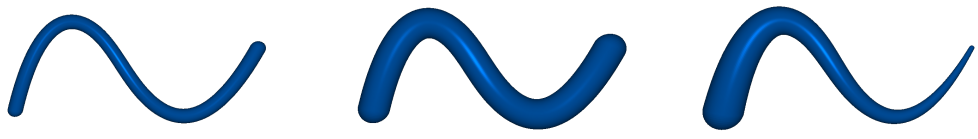
The strain energy function of a curve is often used as an indicator of the fairness of the curve. The strain energy of a curve is defined by the curvature of the curve:

$$EG(\mathbf{c}(t)) = \int_0^1 \|\mathbf{c}''(t)\|^2 dt \tag{13}$$

For BBSC, the strain energy function of BBSC can be defined by borrowing the strain energy function of a curve. BBSC contains two parts: the skeleton line and the radius. In order to consider the effects of the skeleton line and radius on the fairnes at the same time, we consider the control balls as a point in a 4-dimensional Euclidean space and BBSC as a curve in the 4-dimensional space to define its strain energy function.

$$EG(\mathbf{B}(t)) = \int_0^1 \|\mathbf{B}''(t)\|^2 dt = \int_0^1 \left\| \sum_{i=0}^{n} N''_{i,k}(t)\mathbf{Q}_i \right\|^2 dt \tag{14}$$

We illustrate the validity of the BBSC strain energy function by the following example. We show three BBSCs, $\mathbf{B}_1(t), \mathbf{B}_2(t), \mathbf{B}_3(t)$, having the same skeleton line and different radius functions. Among them, $\mathbf{B}_1(t)$ and $\mathbf{B}_2(t)$ have different control ball radius but their radii are constant and have no effect on the strain energy function. Whereas, the control ball radius of $\mathbf{B}_3(t)$ has a variation, which has an effect on the strain energy function, so the strain energy value of $\mathbf{B}_3(t)$ is greater than that of $\mathbf{B}_1(t)$ and $\mathbf{B}_2(t)$. This example shows that the BBSC strain energy function we defined responds to the degree of bending and fairness of BBSC.



(a) $EG(\mathbf{B})_1(t) = 1412856$ (b) $EG(\mathbf{B})_2(t) = 1412856$ (c) $EG(\mathbf{B})_3(t) = 1413273$

Fig. 3. Strain energy of different BBSCs.

5.2 Discrete Expression by Control Balls

The strain energy function of BBSC is defined based on curvature, which presents computational difficulties. The entire geometric characteristics of BBSC are completely defined by the control balls and knot vectors. In this subsection, we convert the continuous expression of the strain energy function based on the BBSC curvature into a discrete expression of the strain energy function based on the control balls and knot vectors to facilitate subsequent calculations.

BBSC is a segmented structure. According to the additivity of the integral, the strain energy function of BSBC can be rewritten in the following form:

$$EG(\mathbf{B}(t)) = \int_0^1 \|\mathbf{B}''(t)\|^2 dt = \sum_{i=k-1}^{n} \int_{t_i}^{t_{i+1}} \left\|\mathbf{B}''_{i-k+1}(t)\right\|^2 dt \tag{15}$$

A common method for calculating B-Spline basis functions is the de-Boor-Cox method, and such an iterative method is too difficult to calculate. We introduce a matrix representation of the B-Spline to transform the iterative form of the BBSC into a non-iterative form.

$$\mathbf{B}_{i-k+1}(t) = \begin{pmatrix} 1 & \frac{t-t_i}{t_{i+1}-t_i} & \frac{(t-t_i)^2}{(t_{i+1}-t_i)^2} & \frac{(t-t_i)^3}{(t_{i+1}-t_i)^3} \end{pmatrix} \cdot \mathbf{M}^4(i) \cdot \begin{pmatrix} \mathbf{Q}_{i-3} & \mathbf{Q}_{i-2} & \mathbf{Q}_{i-1} & \mathbf{Q}_i \end{pmatrix}^{\mathrm{T}} \tag{16}$$

$\mathbf{M}^4(i)$is the core of the B-Spline matrix expression: the coefficient matrix. The second derivative norm of $\mathbf{B}_{i-k+1}(t)$ can be easily derived in the following matrix representation:

$$\mathbf{B}''_{i-k+1}(t) = \begin{pmatrix} 0 & 0 & \frac{2}{(t_{i+1}-t_i)^2} & \frac{6(t-t_i)}{(t_{i+1}-t_i)^3} \end{pmatrix} \cdot \mathbf{M}^4(i) \cdot \begin{pmatrix} \mathbf{Q}_{i-3} & \mathbf{Q}_{i-2} & \mathbf{Q}_{i-1} & \mathbf{Q}_i \end{pmatrix}^{\mathrm{T}} \tag{17}$$

$$\begin{aligned} \|\mathbf{B}''_{i-k+1}(t)\|^2 =& \mathbf{B}''_{i-k+1}(t) \cdot \mathbf{B}''_{i-k+1}(t)^{\mathrm{T}} \\ =& \begin{pmatrix} 0 & 0 & \frac{2}{(t_{i+1}-t_i)^2} & \frac{6(t-t_i)}{(t_{i+1}-t_i)^3} \end{pmatrix} \cdot \mathbf{N}^4(i) \cdot \begin{pmatrix} \mathbf{Q}_{i-3} & \mathbf{Q}_{i-2} & \mathbf{Q}_{i-1} & \mathbf{Q}_i \end{pmatrix}^{\mathrm{T}} \end{aligned} \tag{18}$$

$\mathbf{N}^4(i)$ can be expressed as follows:

$$N^4(i) = M^4(i) \cdot \begin{pmatrix} \mathbf{Q}_{i-3} \cdot \mathbf{Q}_{i-3} & \mathbf{Q}_{i-3} \cdot \mathbf{Q}_{i-2} & \mathbf{Q}_{i-3} \cdot \mathbf{Q}_{i-1} & \mathbf{Q}_{i-3} \cdot \mathbf{Q}_i \\ \mathbf{Q}_{i-2} \cdot \mathbf{Q}_{i-3} & \mathbf{Q}_{i-2} \cdot \mathbf{Q}_{i-2} & \mathbf{Q}_{i-2} \cdot \mathbf{Q}_{i-1} & \mathbf{Q}_{i-2} \cdot \mathbf{Q}_{i-1} \\ \mathbf{Q}_{i-1} \cdot \mathbf{Q}_{i-3} & \mathbf{Q}_{i-1} \cdot \mathbf{Q}_{i-2} & \mathbf{Q}_{i-1} \cdot \mathbf{Q}_{i-1} & \mathbf{Q}_{i-1} \cdot \mathbf{Q}_i \\ \mathbf{Q}_i \cdot \mathbf{Q}_{i-3} & \mathbf{Q}_i \cdot \mathbf{Q}_{i-2} & \mathbf{Q}_i \cdot \mathbf{Q}_{i-1} & \mathbf{Q}_i \cdot \mathbf{Q}_i \end{pmatrix} \cdot \mathbf{M}^4(i)^{\mathrm{T}} \tag{19}$$

Denoting the elements of $\mathbf{N}^4(i)$ as n_{ij}, the polynomial expression of $\|\mathbf{B}''_{i-k+1}(t)\|^2$ is as follows:

$$\left\|\mathbf{B}''_{i-k+1}(t)\right\|^2 = \frac{4n_{22}^i}{(t_{i+1}-t_i)^4} + \frac{12\,(t-t_i)\,\left(n_{23}^i + n_{32}^i\right)}{(t_{i+1}-t_i)^5} + \frac{36(t-t_i)^2 n_{33}^i}{(t_{i+1}-t_i)^6} \tag{20}$$

Integrating Eq. (19), the strain energy function can be obtained as follows

$$EG(\mathbf{B}(t)) = \sum_{i=k-1}^{n} \frac{4n_{22}^i + 6\left(n_{23}^i + n_{32}^i\right) + 12n_{33}^i}{(t_{i+1}-t_i)^3} \tag{21}$$

Denoting the elements of $\mathbf{M}^4(i)$ as m_{ij}, the numerator in Eq. (20) can be derived as follows

$$\begin{aligned} & 4n_{22}^i + 6\left(n_{23}^i + n_{32}^i\right) + 12n_{33}^i \\ &= \left(4m_{20}^i m_{20}^i + 12m_{20}^i m_{30}^i + 12m_{30}^i m_{30}^i\right) \mathbf{Q}_{i-3} \cdot \mathbf{Q}_{i-3} \\ &+ \left(8m_{20}^i m_{21}^i + 12m_{30}^i m_{21}^i + 12m_{31}^i m_{20}^i + 24m_{30}^i m_{31}^i\right) \mathbf{Q}_{i-3} \cdot \mathbf{Q}_{i-2} \\ &+ \left(8m_{20}^i m_{22}^i + 12m_{30}^i m_{22}^i + 12m_{32}^i m_{20}^i + 24m_{30}^i m_{32}^i\right) \mathbf{Q}_{i-3} \cdot \mathbf{Q}_{i-1} \\ &+ \left(8m_{20}^i m_{23}^i + 12m_{30}^i m_{23}^i + 12m_{33}^i m_{20}^i + 24m_{30}^i m_{33}^i\right) \mathbf{Q}_{i-3} \cdot \mathbf{Q}_i \\ &+ \left(4m_{21}^i m_{21}^i + 12m_{31}^i m_{21}^i + 12m_{31}^i m_{31}^i\right) \mathbf{Q}_{i-2} \cdot \mathbf{Q}_{i-2} \\ &+ \left(8m_{21}^i m_{22}^i + 12m_{31}^i m_{22}^i + 12m_{21}^i m_{32}^i + 24m_{31}^i m_{32}^i\right) \mathbf{Q}_{i-2} \cdot \mathbf{Q}_{i-1} \\ &+ \left(8m_{21}^i m_{23}^i + 12m_{23}^i m_{31}^i + 12m_{21}^i m_{33}^i + 24m_{31}^i m_{33}^i\right) \mathbf{Q}_{i-2} \cdot \mathbf{Q}_i \\ &+ \left(4m_{22}^i m_{22}^i + 12m_{22}^i m_{32}^i + 12m_{32}^i m_{32}^i\right) \mathbf{Q}_{i-1} \cdot \mathbf{Q}_{i-1} \\ &+ \left(8m_{22}^i m_{23}^i + 12m_{23}^i m_{32}^i + 12m_{22}^i m_{33}^i + 24m_{32}^i m_{33}^i\right) \mathbf{Q}_{i-1} \cdot \mathbf{Q}_i \\ &+ \left(4m_{23}^i m_{23}^i + 12m_{23}^i m_{33}^i + 12m_{33}^i m_{33}^i\right) \mathbf{Q}_i \cdot \mathbf{Q}_i \end{aligned} \tag{22}$$

Denoting the coefficient of $Q_{i-j} \cdot Q_{i-k}$ as $c_{j,k}^i$, the strain energy function of $\mathbf{B}(t)$ can be written as follows

$$EG(\mathbf{B}(t)) = \sum_{i=3}^{n} \sum_{j=0}^{3} \sum_{k=0}^{j} c_{j,k}^i \mathbf{Q}_{i-j} \cdot \mathbf{Q}_{i-k} \tag{23}$$

Eq. (22) is the discrete expression of the BBSC strain energy function about the control balls and the knot vector.

6 BLENDING OF BALL B-SPLINE CURVE

In this section, we will introduce the specific method for solving blending curves. First, we will introduce how the control balls of the blending curve is constrained by the continuity conditions. Secondly, we will present the specific optimization process for the BBSC strain energy function. Finally, we will give the general algorithmic flow of BBSC blending problem. We use $\mathbf{C}_1(t), \mathbf{C}_2(t)$ to denote the original BBSCs, and $\mathbf{B}(t)$ to denote the blending BBSC.

6.1 Continuity Conditions

In the BBSC blending task, the blending BBSC needs to satisfy G^2 continuous conditions with the original BBSCs at the junction. Naturally, we translate the constraints for blending BBSC into constraints for the control balls.

We use the BBSC matrix expression of Equation(16) to convert $\mathbf{B}(t)$ into the form of control balls expression. The specific expressions of the first three control balls of blending BBSC and the last three control balls with respect to the G^2 continuous degree of freedom parameters can be obtained as follows

$$\begin{cases} \mathbf{Q}_0 = \mathbf{C}_1(1) \\ \mathbf{Q}_1 = \alpha_1 \frac{(t_k - t_1)}{3} \mathbf{C}_1'(1) + \mathbf{C}_1(1) \\ \mathbf{Q}_2 = \alpha_1^2 \frac{(t_k - t_2)(t_{k+1} - t_2)}{6} \mathbf{C}_1''(1) + \alpha_1 \frac{t_{k+1} + t_k - t_2 - t_1}{3} \mathbf{C}_1'(1) + \beta_1 \frac{(t_k - t_2)(t_{k+1} - t_2)}{6} \mathbf{C}_1'(1) + \mathbf{C}_1(1) \\ \mathbf{Q}_{n-2} = \alpha_2^2 \frac{(t_{n+2} - t_n)(t_{n+2} - t_{n-1})}{6} \mathbf{C}_2''(0) + \alpha_2 \frac{t_n + t_{n+1} - t_{n+3} - t_{n+2}}{3} \mathbf{C}_2'(1) + \beta_2 \frac{(t_{n+2} - t_n)(t_{n+2} - t_{n-1})}{6} \mathbf{C}_2'(0) + \mathbf{C}_2(0) \\ \mathbf{Q}_{n-1} = \alpha_2 \frac{(t_n - t_{n+3})}{3} \mathbf{C}_2'(0) + \mathbf{C}_2(0) \\ \mathbf{Q}_n = \mathbf{C}_2(0) \end{cases} \tag{24}$$

6.2 Two-Step Optimization Process

The optimization method is divided into two main steps: preprocessing and constrained optimization based on the trust region. In the preprocessing section, we transform the BBSC strain energy function from the form based on the control balls expression to the form based on the continuity condition parameters by the continuity conditions of BBSCs and the control balls optimality condition. In the constrained optimization section, we introduce the trust-region-reflection algorithm and give the specific form of the optimization problem with the constraint conditions.

6.2.1 Preprocess Solving. When $n = 5$, the blending curve has 6 control balls, all control balls can be transformed into expressions of by the continuity condition parameters. To obtain the optimal BBSC control balls for the blending curve, we need only to optimally solve for 4 parameters $\alpha_1, \alpha_2, \beta_1, \beta_2$ and then bring parameters into the control ball expression.

Note that in the optimization process, it is necessary to ensure that the control ball radius is greater than 0, α_1 and α_2 are not less than 0. The optimization problem is expressed as follows:

$$\begin{array}{cc} \min & EG_{\mathbf{B}(t)}\ (\alpha_1, \beta_1, \alpha_2, \beta_2) \\ \text{s.t.} & -\left(a_i \alpha_1^2 + b_i \alpha_1 + c_i \beta_1 + d_i - \epsilon\right) \le 0, i = 1, 2 \\ & -\left(a_i \alpha_2^2 + b_i \alpha_2 + c_i \beta_2 + d_i - \epsilon\right) \le 0, i = n-2, n-1 \\ & \alpha_1 \ge 0, \alpha_2 \ge 0 \end{array} \tag{25}$$

When $n > 5$, the blending BBSC has more than 6 control balls, where Q_3 Q_{n-3} are not controlled by the continuity constraint. In this case, the optimization problem is expressed as follows:

$$\begin{array}{cc} \min & EG_{\mathbf{B}(t)}\ (\alpha_1, \beta_1, \alpha_2, \beta_2, \mathbf{Q}_3, \ldots, \mathbf{Q}_{n-3}) \\ \text{s.t.} & -\left(a_i\alpha_1^2 + b_i\alpha_1 + c_i\beta_1 + d_i - \epsilon\right) \le 0, i = 1, 2 \\ & -\left(a_i\alpha_2^2 + b_i\alpha_2 + c_i\beta_2 + d_i - \epsilon\right) \le 0, i = n-2, n-1 \\ & \alpha_1 \ge 0, \alpha_2 \ge 0 \end{array} \tag{26}$$

Considering that as the number of BBSC segments increases, i.e., n increases, more and more control balls need to be optimized, making the optimization problem difficult due to too many optimization parameters. We first optimize the control balls by transforming them into the expression about the continuity parameter to reduce the number of parameters to be optimized, thus simplifying the optimization problem.

$$\frac{\partial EG_{\mathbf{B}(t)}}{\partial \mathbf{Q}_i} = \mathbf{0}, \quad i = 3, \ldots, n-3 \tag{27}$$

After transforming all control balls into the expression based on $\alpha_1, \alpha_2, \beta_1, \beta_2$, the strain energy function of BBSC can be written in the following form:

$$\begin{array}{l} EG_{\mathbf{B}}(t)\ (\alpha_1, \alpha_2, \beta_1, \beta_2) \\ = \lambda_0\alpha_1^4 + \lambda_1\alpha_1^3 + \lambda_2\alpha_1^2 + \lambda_3\alpha_1 \\ +\lambda_4\alpha_2^4 + \lambda_5\alpha_2^3 + \lambda_6\alpha_2^2 + \lambda_7\alpha_2 \\ +\lambda_8\alpha_1^2\alpha_2^2 + \lambda_9\alpha_1^2\alpha_2 + \lambda_{10}\alpha_1\alpha_2^2 + \lambda_{11}\alpha_1\alpha_2 \\ +\lambda_{12}\alpha_1^2\beta_1 + \lambda_{13}\alpha_1\beta_1 + \lambda_{14}\alpha_2^2\beta_1 + \lambda_{15}\alpha_2\beta_1 \\ +\lambda_{16}\alpha_1^2\beta_2 + \lambda_{17}\alpha_1\beta_2 + \lambda_{18}\alpha_2^2\beta_2 + \lambda_{19}\alpha_2\beta_2 \\ +\lambda_{20}\beta_1^2 + \lambda_{21}\beta_2^2 + \lambda_{22}\beta_1\beta_2 + \lambda_{23}\beta_1 + \lambda_{24}\beta_2 + \lambda_{25} \end{array} \tag{28}$$

Next, we describe how to determine the coefficients $\lambda_i (i = 0, ..., 25)$. All of the control balls of $\mathbf{B}(t)$ can be expressed by $\alpha_1, \alpha_2, \beta_1, \beta_2$ in the following form:

$$\mathbf{Q}_i = \mathbf{A}_i\alpha_1^2 + \mathbf{B}_i\alpha_2^2 + \mathbf{C}_i\alpha_1 + \mathbf{D}_i\alpha_2 + \mathbf{E}_i\beta_1 + \mathbf{F}_i\beta_2 + \mathbf{G}_i \tag{29}$$

where $\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i, \mathbf{D}_i, \mathbf{E}_i, \mathbf{F}_i, \mathbf{G}_i$ are all four-dimensional vectors. $\mathbf{Q}_i$ can be written as a vector:

$$\begin{pmatrix} \mathbf{A}_i & \mathbf{B}_i & \mathbf{C}_i & \mathbf{D}_i & \mathbf{E}_i & \mathbf{F}_i & \mathbf{G}_i \end{pmatrix} \cdot \begin{pmatrix} \alpha_1^2 & \alpha_2^2 & \alpha_1 & \alpha_2 & \beta_1 & \beta_2 & 1 \end{pmatrix}^{\mathbf{T}} \tag{30}$$

Thus, when we find the dot product for any two control balls $\mathbf{Q}_i$ and $\mathbf{Q}_j$, we have the following result:

$$\begin{array}{l} \mathbf{Q}_i \cdot \mathbf{Q}_j = \\ +\mathbf{A}_i \cdot \mathbf{A}_j\alpha_1^4 + \left(\mathbf{A}_i \cdot \mathbf{C}_j + \mathbf{C}_i \cdot \mathbf{A}_j\right)\alpha_1^3 \\ + \left(\mathbf{A}_i \cdot \mathbf{G}_j + \mathbf{G}_i \cdot \mathbf{A}_j + \mathbf{C}_i \cdot \mathbf{C}_j\right)\alpha_1^2 \\ +\mathbf{B}_i \cdot \mathbf{B}_j\alpha_2^4 + \left(\mathbf{B}_i \cdot \mathbf{D}_j + \mathbf{D}_i \cdot \mathbf{B}_j\right)\alpha_2^3 \\ + \left(\mathbf{B}_i \cdot \mathbf{G}_j + \mathbf{G}_i \cdot \mathbf{B}_j + \mathbf{D}_i \cdot \mathbf{D}_j\right)\alpha_2^2 \\ + \left(\mathbf{C}_i \cdot \mathbf{G}_j + \mathbf{G}_i \cdot \mathbf{C}_j\right)\alpha_1 + \left(\mathbf{D}_i \cdot \mathbf{G}_j + \mathbf{G}_i \cdot \mathbf{D}_j\right)\alpha_2 \\ + \left(\mathbf{A}_i \cdot \mathbf{B}_j + \mathbf{B}_i \cdot \mathbf{A}_j\right)\alpha_1^2\alpha_2^2 + \left(\mathbf{A}_i \cdot \mathbf{D}_j + \mathbf{D}_i \cdot \mathbf{A}_j\right)\alpha_1^2\alpha_2 \\ + \left(\mathbf{C}_i \cdot \mathbf{B}_j + \mathbf{B}_i \cdot \mathbf{C}_j\right)\alpha_1\alpha_2^2 + \left(\mathbf{C}_i \cdot \mathbf{D}_j + \mathbf{D}_i \cdot \mathbf{C}_j\right)\alpha_1\alpha_2 \\ + \left(\mathbf{A}_i \cdot \mathbf{E}_j + \mathbf{E}_i \cdot \mathbf{A}_j\right)\alpha_1^2\beta_1 + \left(\mathbf{C}_i \cdot \mathbf{E}_j + \mathbf{E}_i \cdot \mathbf{C}_j\right)\alpha_1\beta_1 \\ + \left(\mathbf{B}_i \cdot \mathbf{E}_j + \mathbf{E}_i \cdot \mathbf{B}_j\right)\alpha_2^2\beta_1 + \left(\mathbf{D}_i \cdot \mathbf{E}_j + \mathbf{E}_i \cdot \mathbf{D}_j\right)\alpha_2\beta_1 \\ + \left(\mathbf{A}_i \cdot \mathbf{F}_j + \mathbf{F}_i \cdot \mathbf{A}_j\right)\alpha_1^2\beta_2 + \left(\mathbf{C}_i \cdot \mathbf{F}_j + \mathbf{F}_i \cdot \mathbf{C}_j\right)\alpha_1\beta_2 \\ + \left(\mathbf{B}_i \cdot \mathbf{F}_j + \mathbf{F}_i \cdot \mathbf{B}_j\right)\alpha_2^2\beta_2 + \left(\mathbf{D}_i \cdot \mathbf{F}_j + \mathbf{F}_i \cdot \mathbf{D}_j\right)\alpha_2\beta_2 \\ +\mathbf{E}_i \cdot \mathbf{E}_j\beta_1^2 + \mathbf{F}_i \cdot \mathbf{F}_j\beta_2^2 + \left(\mathbf{E}_i \cdot \mathbf{G}_j + \mathbf{G}_i \cdot \mathbf{E}_j\right)\beta_1 \\ + \left(\mathbf{F}_i \cdot \mathbf{G}_j + \mathbf{G}_i \cdot \mathbf{F}_j\right)\beta_2 + \left(\mathbf{F}_i \cdot \mathbf{E}_j + \mathbf{E}_i \cdot \mathbf{F}_j\right)\beta_1\beta_2 \\ +\mathbf{G}_i \cdot \mathbf{G}_j \end{array} \tag{31}$$

Combining Equations (28) and (31), the specific form of each coefficient λ in Equation (28) can be obtained.

6.2.2 Constrained Optimization. In the previous subsection, we obtained the expression form of the BBSC strain energy function with $\alpha_1, \alpha_2, \beta_1, \beta_2$. Its specific form is Equation (28).

When optimizing the strain energy function, it is important to note that the radius of the control ball in BBSC must be positive. Therefore, we need to perform a constrained optimization of the strain energy function with the constraint that the control ball radius is greater than 0. Our optimization problem can be expressed in the following form:

$$\begin{array}{cc} \min & EG_{\mathbf{B}(t)}\left(\alpha_1, \beta_1, \alpha_2, \beta_2\right) \\ \text{s.t.} & -\left(a_i\alpha_1^2 + b_i\alpha_1 + c_i\beta_1 + d_i - \epsilon\right) \leq 0, i = 1, 2 \\ & -\left(a_i\alpha_2^2 + b_i\alpha_2 + c_i\beta_2 + d_i - \epsilon\right) \leq 0, i = n-2, n-1 \\ & \alpha_1 \geq 0, \alpha_2 \geq 0 \end{array} \tag{32}$$

Specifically, we perform constrained optimization of the strain energy function in MATLAB.We use *fmincon* in *Optimization Toolbox*.

6.3 Algorithm Process

In this section, we describe the whole algorithm process. We start from the minimum number of control balls required for the blending curve, i.e., $n = 6$, and keep increasing the number of control balls until the strain energy of the curve no longer decreases at a certain number of control balls, at which point the blending curve is taken as the optimized result. We decribe the algorithm flow in detail in Algorithm 1.

As the number of control balls increases, the knot vector of ball B-Spline curves also changes. We use the knot vectors of quasi-uniform B-Spline curves as the knot vectors of blending curves. It is constructed as

$$\{\underbrace{0, \ldots, 0}_{k}, t_k, \ldots, t_n, \underbrace{1, \ldots, 1}_{k}\} \tag{33}$$

$$t_i = \frac{i-k+1}{n-k+2} \tag{34}$$

7 EXPERIMENTS

In this section, we conduct experiments to demonstrate our approach. First, we mainly compare with Jiang's method to show that our method can obtain blending BBSC with better fairness. Second, we use our method to fill in the missing vessel Branches data to show the performance of our method in practical applications.

In terms of evaluation metrics for the comparison experiments, in addition to strain energy, we also compared the winding number. Strain energy and winding number are commonly used evaluation metrics in previous studies. In our work, strain energy is the optimization target. winding number describes the angular change of the curve tangent as it moves along the curve. For a planar curve $c(t)$ with curvature $k(t)$, its winding number W can be calculated as

$$W = \frac{1}{2\pi}\int_0^1 |k(t)|\,\|c'(t)\|\,dt \tag{35}$$

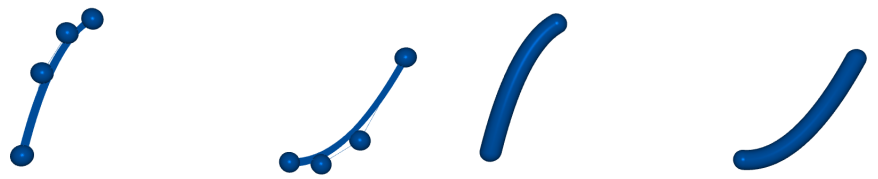
7.1 Blending of the Generated BBSCs

In the experiments of this section, we still use the strain energy of BBSC and the winding number of the skeleton line as evaluation metrics. Then our algorithm is compared experimentally with Jiang's method[Jiang et al. 2014].

Algorithm 1 BBSC Blending Algorithm

Require: 2 BBSCs $\mathbf{C}_1(t)$ and $\mathbf{C}_2(t)$
Ensure: A blending BBSC $\mathbf{B}(t)$ with minimum strain energy

```
Initialize:n = 5, EG_min = +∞
loop
    Determine the knot vector of the blending curve according to n
    Express the control balls Q_i(i = 0, 1, 2, n − 2, n − 1, n) as an expression shaped like (30) according to equation (24)
    Solve the equation (27) to express the control balls Q_i(i = 3, ..., n−3) as an expression shaped like (30)
    Determine the objective function according to equation (28)
    Minimize the energy function using fmincon in Optimization Toolbox
    Substitute the optimal α_1, α_2, β_1, β_2 into the expression of the control balls
    Calculate the strain energy according to equation (23)
    if EG_min > EG then
        Build the B(t) according to the knot vector and control balls
        EG_min = EG, n = n + 1
    else
        The last B(t) is the result BBSC
        break loop
    end if
end loop
```



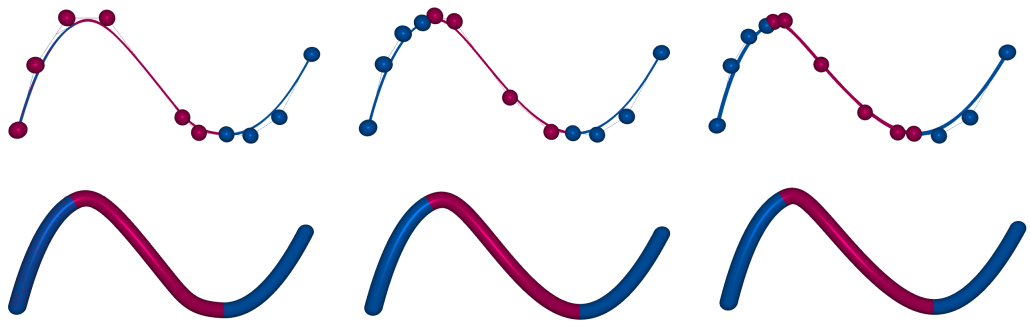
(a) Skeletons of input BBSCs. (b) Input BBSCs.

Fig. 4. Input of BBSC blending.

Table 1. Numerical comparison of blending generated BBSCs.

Metrics	Jiang et al.[Jiang et al. 2014]	3 segments.(Ours)	4 segments.(Ours)
strain energy	59644.50	36318.71	34561.00
winding number	0.359	0.357	0.347

In the above experiments, we can find that the BBSC obtained by our method has smaller strain energy and winding number compared with Jiang's method[Jiang et al. 2014]. in addition, we find that the stain energy and winding number of BBSC become smaller as the number of segments increases. This indicates that increasing the number of BBSC segments and increasing the solution space lead to better fairness of the obtained BBSC.



(a) Jiang et al. (b) 3 segments. (c) 4 segment.

Fig. 5. Comparison of Jiang's method to our one using 3 and 4 segments. Jiang's algorithm introduces visible G^2 continuity, we obtain a smoother solution with better fairness.

Table 2. Numerical comparison of examples of blending BBSCs.

Examples.	strain energy		winding number	
	Jiang et al.[Jiang et al. 2014]	Ours.	Jiang et al.[Jiang et al. 2014]	Ours.
Example 1.	4534.75	1914.72	0.249	0.250
Example 2.	4244.48	2133.34	0.316	0.2501
Example 3.	8161.78	5694.90	0.304	0.304
Example 4.	4530.97	2912.06	0.410	0.412

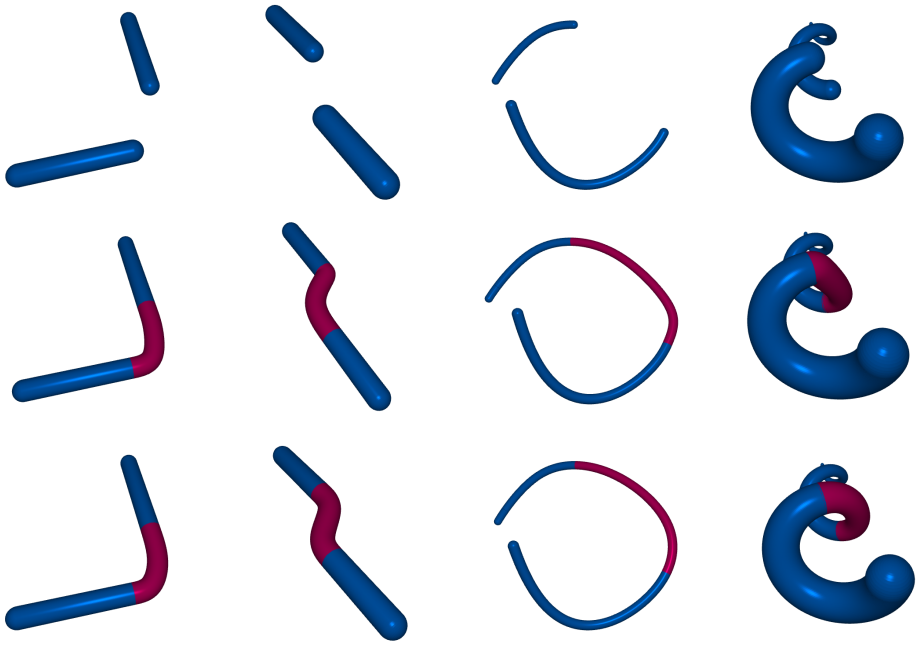
We also performed comparative experiments on other samples. In Fig. 6, the first row shows the original input data; the second row shows the experimental results obtained by Jiang's method[Jiang et al. 2014]; and the last row shows the experimental results given by our method. Table 2 shows the numerical comparison results of Fig. 6. This result further verifies that our blending method is able to obtain blending results with better fairness compared with other methods.

7.2 Blending of Vessel Branches

There is substantial evidence that the geometric features of vessels have a strong influence on hemodynamics[Del et al. 2015] and consequently on the development of vascular diseases, such as atherosclerosis, cerebral aneurysmal disease, etc[Piccinelli et al. 2009; Yonezawa et al. 2002]. The geometric features of vessels are based on their geometric models. However, due to the limitations of segmentation algorithms[Florez-Valencia et al. 2010; Wen et al. 2015], complete volume data are often difficult to obtain, and the vascular models generated with these data often show various gaps. In this section, we show that the modeling algorithm for BBSCs is useful in the field of medical imaging. We first use the BBSC to model the cerebral vascular data obtained by MRA. Due to the incompleteness of the original data, the obtained BBSC model has some residuals. We will use the blending algorithm of the BBSC to repair this fragmented cerebral vascular model.

8 CONCLUSION

In this paper, a new BBSC blending algorithm is proposed. The method uses a ball B-Spline curve to model the blending curve. Compared with the method of modeling blending curves using ball



(a) Example 1. (b) Example 2. (c) Example 3. (d) Example 4.

Fig. 6. Comparison of Jiang's method to our method on more examples. The first row shows the original input data; the second row shows the experimental results obtained by Jiang's method[Jiang et al. 2014]; and the last row shows the experimental results given by our method. Our method obtain a smoother solution with better fairness.

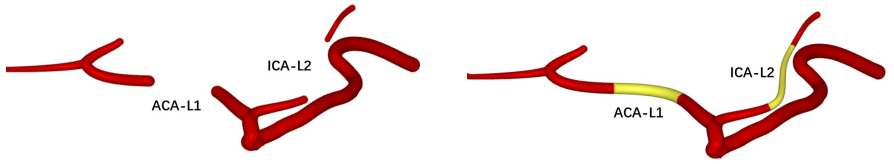


Fig. 7. Examples of vessel branches repairing.

Table 3. Numerical comparison of examples of vessel branches repairing.

Vessel Branch	Number of Segments	metrics		time
		strain energy	winding number	
ICA-L2	3	1675.04	0.27	22.67ms
	4	1490.67	0.25	36.12ms
ACA-L1	3	372.71	0.19	36.97ms
	4	368.25	0.19	44.14ms

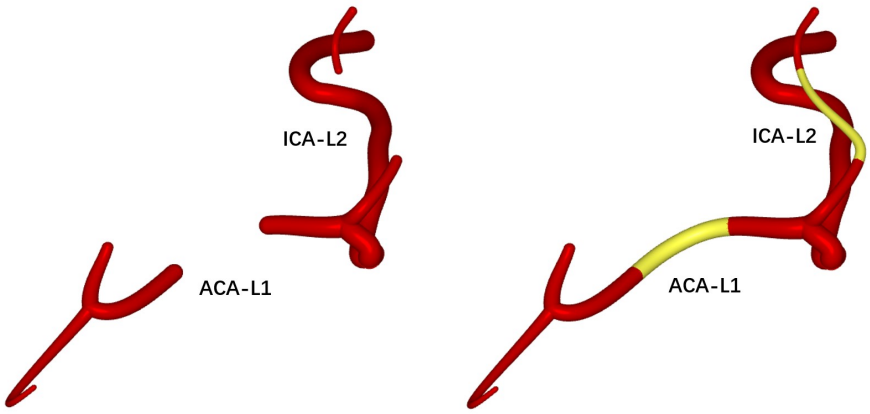


Fig. 8. Examples of vessel branches repairing.

Bézier curves, our method provides more degrees of freedom and expands the solution space to obtain blending BBSCs with better smoothness.

In addition, we consider both the skeleton line and radius of the BBSC, and treat the control ball as a point in the four-dimensional space. Compared with considering the skeleton line and radius of BBSC separately, our method can find the skeleton line and radius of the optimal blending BBSC in the same space, keeping the consistency of both.

In terms of computation, in order to avoid the problem of too large amount of parameters due to expanding the solution space, we propose a two-step optimization process that limits the number of optimization parameters to 4, thus keeping our method real-time.

Finally, we experimentally demonstrate the superiority of our method: increasing the degrees of freedom and expanding the solution space has a significant effect on obtaining a better blending BBSC; at the same time, considering the skeleton line and radius of BBSC can keep the consistency of both well, thus obtaining a blending BBSC with better smoothness; the method has a good computational performance and can keep real-time during the computation.

ACKNOWLEDGMENTS

The authors want to thank the anonymous reviewers for their constructive comments. This research was partially supported by the Beijing Municipal Science and Technology Commission and Zhongguancun Science Park Management Committee (No.Z221100002722020),National Nature Science Foundation of China (No.61972041,No. 62072045), National Key R&D Program of China (No.2020YFC1523300), and Innovation &Transfer Fund of Peking University Third HospitalNo.BYSYZHKC2021110

REFERENCES

X. Ao, Z. Wu, and M. Zhou. 2009. "Real Time Animation of Trees Based on BBSC in Computer Games." In: *Hindawi Publishing Corporation*, 5.

Bachir Belkhatir, Driss Sbibih, and Ahmeh Zidna. 2008. "G1 Blending B-Spline Surfaces and Optimization." In: *Modelling, Computation and Optimization in Information Systems and Management Sciences*. Ed. by Hoai An Le Thi, Pascal Bouvry, and Tao Pham Dinh. Springer Berlin Heidelberg, Berlin, Heidelberg, 458–467. ISBN: 978-3-540-87477-5.

Del et al.. 2015. "Incompleteness of the Circle of Willis Correlates Poorly with Imaging Evidence of Small Vessel Disease. A Population-based Study in Rural Ecuador (the Atahualpa Project)." *Journal of stroke and cerebrovascular diseases: The official journal of National Stroke Association*, 24, 1, 73–77.

L. Florez-Valencia, J. Azencot, and M. Orkisz. 2010. "Algorithm for blood-vessel segmentation in 3D images based on a right generalized cylinder model: application to carotid arteries." In: *International Conference on Computer Vision Graphics*.

Erich Hartmann. 2001. "Parametric G n blending of curves and surfaces." *The Visual Computer*.

Q. Jiang, Z. Wu, T. Zhang, X. Wang, and H. S. Seah. 2014. "G2-Continuity Blending of Ball B-Spline Curve Using Extension." In: *International Conference on Computer-aided Design Computer Graphics*.

X. Liu, X. Wang, Z. Wu, D. Zhang, and X. Liu. 2020. "Extending Ball B-spline by B-spline." *Computer Aided Geometric Design*, 82, 101926.

Y. J. Liu1 et al.. 2009. "NURBS curve blending using extension." *Journal of Zhejiang University(Science A:An International Applied Physics Engineering Journal)*.

M. Piccinelli, A. Veneziani, D. A. Steinman, A. Remuzzi, and L. Antiga. 2009. "A Framework for Geometric Analysis of Vascular Structures: Application to Cerebral Aneurysms." *IEEE Transactions on Medical Imaging*, 28, 8, 1141.

H. S. Seah. 2007. "Skeleton Based Parametric Solid Models: Ball B-Spline Curves." In: *IEEE International Conference on Computer-aided Design Computer Graphics*.

H. S. Seah and Z. Wu. 2005. "Ball B-Spline Based Geometric Models in Distributed Virtual Environments."

Y. Tang, Z. Wu, and M. Zhou. 2009. "Sketching 3D Plant Based on Ball B-Spline Curves and L-system." In: *Plant Growth Modeling, Simulation, Visualization, Applications*.

X. Wang, E. Liu, Z. Wu, F. Zhai, and YC Zhu.... 2016. "Skeleton-based cerebrovascular quantitative analysis." *Bmc Medical Imaging*, 16, 1.

X. Wang, Z. Wu, J. Shen, T. Zhang, X. Mou, and M. Zhou. 2016. "Repairing the cerebral vascular through blending Ball B-Spline curves with G2 continuity." *Neurocomputing*, 173, JAN.15PT.3, 768–777.

L. Wen, X. Wang, Z. Wu, M. Zhou, and J. S. Jin. 2015. "A novel statistical cerebrovascular segmentation algorithm with particle swarm optimization." *Neurocomputing*, 148, 569–577.

Z. Wu, M. Zhou, and X. Wang. 2009. "Interactive Modeling of 3D Tree with Ball B-Spline Curves." *International Journal of Virtual Reality*.

Z. Wu, M. Zhou, X. Wang, X. Ao, and R. Song. 2006. "An Interactive System of Modeling 3D Trees with Ball B-Spline Curves." In: *International Symposium on Plant Growth Modeling Applications*.

X. Xu, C. Leng, and Z. Wu. 2011. "Rapid 3D Human Modeling and Animation Based on Sketch and Motion Database." In: *Workshop on Digital Media Digital Content Management*.

L. U. Yawen. 2011. "Algorithm of G 2 blending NURBS surfaces based on adjustable parameters." *Computer Engineering and Applications*.

T. Yonezawa, H. Mizuno, and M. Abe. 2002. "Robustness of Automatic Labeling with HMM Phoneme Models." *Ieice Technical Report Speech*, 102, 292, 17–22.

L. Zhao, S. Lu, X. Guo, W. Wen, and W. Sheng. 2011. "3D Shape Reconstruction and Realistic Rendering of Flowering Rape (Brassica napus L.)" In: *Multimedia and Signal Processing (CMSP), 2011 International Conference on*.

Shifeng Zhao, Zhongke Wu, and Mingquan Zhou. 2010. "A multi-scale method for extraction of cerebral blood vessles." In: *2010 IEEE International Conference on Progress in Informatics and Computing*. Vol. 2, 1280–1283. DOI: 10.1109/PIC.2010.5687927.

T. Zhu, T. Feng, Z. Yan, H. S. Seah, and X. Yan. 2008. "Plant Modeling Based on 3D Reconstruction and Its Application in Digital Museum." *Ijvr*.