

# Variational Ball Skinning

Greg Slabaugh<sup>1</sup>, Gozde Unal<sup>2</sup>, Tong Fang<sup>1</sup>, Jarek Rossignac<sup>3</sup>, Brian Whited<sup>3</sup>

<sup>1</sup> Siemens Corporate Research, Princeton NJ 08540 USA

<sup>2</sup> Sabanci University, Istanbul Turkey

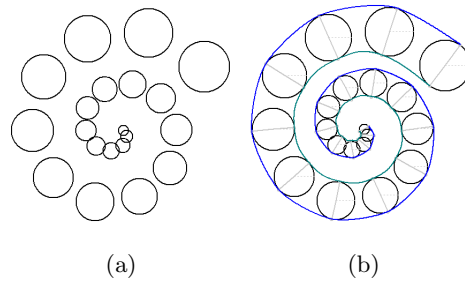
<sup>3</sup> Georgia Institute of Technology, Atlanta, GA 30332 USA

**Abstract.** This paper considers the problem of computing an interpolating envelope of an ordered set of 2D balls. By construction, the envelope is constrained to be  $C^1$  continuous, and for each ball, it touches the ball at a point and is tangent to the ball at the point of contact. Using an energy formulation, we derive differential equations that are designed to minimize the envelope’s arc length and/or curvature subject to these constraints. Given an initial envelope, we update the envelope’s parameters using the differential equations until convergence occurs. We demonstrate the method’s usefulness in generating interpolating envelopes of balls of different sizes and in various configurations.

## 1 Introduction

In this paper, we consider the geometric problem of *ball skinning*, which we define to be the computation of an interpolating envelope of a set of balls; an example appears in Figure 1. This problem arises in numerous applications, including character skinning, molecular surface model generation, and modeling of tubular structures. The balls can have different radii, be configured in different positions, and may or may not overlap. In our formulation of the problem, we require that the envelope touch each ball at a point of contact, and that the envelope be tangent to the ball at the point of contact. The envelope then forms a “skin” that rests on and interpolates the underlying balls.

For a given configuration of balls, there exist an infinite number of possible solutions to this problem as expressed above. To formulate the problem so that it is well-posed, we seek the envelope that has minimal arc length and/or curvature. We achieve this by deriving differential equations that minimize an energy based on this constrained variational problem. Given an initial envelope, we evolve the envelope’s parameters using the differential equations until convergence occurs. This produces the envelope that has minimal length and/or curvature, touches each ball at a point of contact, and is tangent to the ball at the point of contact. In this sense, the method provides an optimal constrained interpolation of the balls. In this paper we consider both *one-sided* and *two-sided* envelopes. A one-sided envelope is a contour that rests on one side of a collection of balls such as that portrayed in Figure 2 (a), while a two-sided envelope defines an interpolating region that has an inside and outside, as demonstrated in Figure 1.



**Fig. 1.** An example ball skinning. Given an ordered sequence of balls (a), we produce an envelope that optimally interpolates the balls (b). This envelope consists of two splines (green and blue) and is computed using differential equations.

### 1.1 Related Work

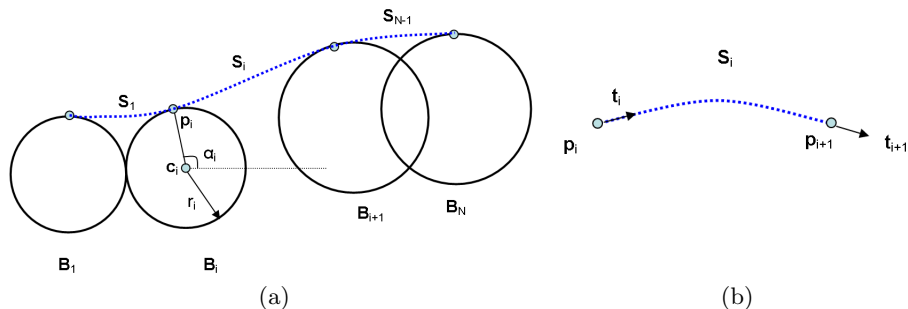
The problem of skinning appears in various contexts. In computer graphics and animation, often an articulated object or character is constructed by forming a layered representation consisting of a skeletal structure and a corresponding geometric skin [1]. The skeleton has fewer degrees of freedom and is simpler to adjust by an animator. Given a new skeletal pose, the skinning algorithm is responsible for deforming the geometric skin to respond to the motion of the underlying skeleton.

The problem of ball skinning appears frequently in the context of computational chemistry and molecular biology, when generating surface meshes for molecular models [2] [3] [4]. Several algorithms exist to skin a molecular model to produce a  $C^1$  continuous surface that is tangent smooth and has high mesh quality. These methods are typically either based on Delaunay triangulation [2] or by finding the isosurface of an implicit function [4]. The work of [4] derives a special subset of envelope surfaces that is piece-wise quadratic. While the surfaces generated by these methods are tangent to the balls and have smoothness at the point of tangency, none of these existing methods provide an optimally smooth envelope, unlike the method we present here.

In our application, we are interested in modeling the geometry of a blood vessel that has been identified using a 2D variant of Pearling [5], a ball packing algorithm that places numerous balls of different radii so that they fit snugly inside an imaged blood vessel. Given these balls, we would like to find an smooth,  $C^1$  envelope that smoothly interpolates the balls. This surface can then be used for visualization of the blood vessel as well as measurements such as volume or surface area.

### 1.2 Our Contribution

We model the envelope as a  $C^1$  spline, which, by construction, must touch each ball at a point of contact and be tangent to the ball at the point of contact. We



**Fig. 2.** Depiction of the problem. In (a), we would like to find the envelope (dotted curve) that interpolates the ordered set of balls. In (b), we show a depiction of a segment of the envelope.

then provide two novel derivations, one for deforming this constrained spline to minimize its arc length; and a second derivation for minimizing its curvature. The result of these derivations are differential equations, which we then solve to update a given spline to its optimal position. We then show experimental examples of how these differential equations are used to perform *optimally smooth* skinning of balls.

## 2 Envelope representation

In this paper we consider the problem of *smoothly* interpolating between a set of touching balls. Our objective is to find an envelope  $S$ , that satisfies several geometric criteria:

1. The envelope should be modeled by a point of contact with a ball.
2. The envelope should be tangent to a ball at the point of contact.
3. The envelope should optimize an energy functional composed of arc length and/or curvature terms.

A depiction of the problem is presented in Figure 2 (a). The envelope we would like to determine is depicted as a dotted line, and is a  $C^1$  curve passing through the point of contact on each ball, and at each contact point, the envelope is tangential to the ball.

The envelope  $S$  is composed of a set of envelope segments,  $S_i$ , for  $i = 1 \dots N-1$ , where  $N$  is the total number of balls (and  $N-1$  is the total number of segments).

### 2.1 Segment representation

As shown in Figure 2 (b), we choose to model each segment  $S_i$  using a spline that starts at point  $p_i$  in direction  $t_i$ , and ends at point  $p_{i+1}$  in direction  $t_{i+1}$ .

We model the segment using a cubic polynomial curve

$$\mathbf{S}_i = \mathbf{A}_i t^3 + \mathbf{B}_i t^2 + \mathbf{C}_i t + \mathbf{D}_i, \quad (1)$$

since the four constraints require four degrees of freedom. For the  $i$ th segment,  $\mathbf{A}_i$ ,  $\mathbf{B}_i$ ,  $\mathbf{C}_i$ , and  $\mathbf{D}_i$  are constants, and  $t \in [0, 1]$  is a time variable that parameterizes the curve.

To determine the constants for a segment, we must satisfy the constraints,  $\mathbf{S}_i|_{t=0} = \mathbf{p}_i$ ,  $\frac{d\mathbf{S}_i}{dt}|_{t=0} = \mathbf{t}_i$ ,  $\mathbf{S}_i|_{t=1} = \mathbf{p}_{i+1}$ , and  $\frac{d\mathbf{S}_i}{dt}|_{t=1} = \mathbf{t}_{i+1}$ . With these constraints, and the derivative of the segment,

$$\mathbf{S}'_i = \frac{d\mathbf{S}_i}{dt} = 3\mathbf{A}_i t^2 + 2\mathbf{B}_i t + \mathbf{C}_i, \quad (2)$$

we obtain a system of four equations for the four constants:  $\mathbf{D}_i = \mathbf{p}_i$ ,  $\mathbf{C}_i = \mathbf{t}_i$ ,  $\mathbf{A}_i + \mathbf{B}_i + \mathbf{C}_i + \mathbf{D}_i = \mathbf{p}_{i+1}$ , and  $3\mathbf{A}_i + 2\mathbf{B}_i + \mathbf{C}_i = \mathbf{t}_{i+1}$ , which is easily solved, yielding

$$\begin{aligned} \mathbf{A}_i &= -2\mathbf{p}_{i+1} + 2\mathbf{p}_i + \mathbf{t}_i + \mathbf{t}_{i+1} \\ \mathbf{B}_i &= 3\mathbf{p}_{i+1} - 3\mathbf{p}_i - 2\mathbf{t}_i - \mathbf{t}_{i+1} \\ \mathbf{C}_i &= \mathbf{t}_i \\ \mathbf{D}_i &= \mathbf{p}_i \end{aligned} \quad (3)$$

## 2.2 Endpoints

We now have a way to model each segment of the envelope. But we have not yet described how to determine the endpoints  $\mathbf{p}_i$ ,  $\mathbf{p}_{i+1}$  and their respective tangents,  $\mathbf{t}_i$ ,  $\mathbf{t}_{i+1}$  of each segment. As shown in Figure 2 (a), we can represent the point of contact  $\mathbf{p}_i$  on the  $i$ th ball as

$$\mathbf{p}_i = \mathbf{c}_i + \begin{bmatrix} r_i \cos \alpha_i \\ r_i \sin \alpha_i \end{bmatrix}, \quad (4)$$

where  $r_i$  is the radius of ball,  $\mathbf{c}_i$  is its center, and  $\alpha_i$  is an angle. In addition, we can represent the tangent

$$\mathbf{t}_i = \begin{bmatrix} -a_i \sin \alpha_i \\ a_i \cos \alpha_i \end{bmatrix}, \quad (5)$$

where  $a_i$  is a stiffness coefficient that controls the influence of the tangential constraint. Each  $a_i$  is fixed to be half the distance between the next and previous ball centers (for the first and last balls, it is the distance between the ball center and its neighbor ball center). Note that both the point  $\mathbf{p}_i$  and the tangent  $\mathbf{t}_i$  are only a function of the angle  $\alpha_i$ , since the radius, center, and stiffness coefficient of the ball are fixed.

We now have a way to represent the envelope  $\mathbf{S}$  as a set of segments  $\mathbf{S}_i$ , where each segment  $\mathbf{S}_i$  interpolates between the points of contact  $\mathbf{p}_i$ ,  $\mathbf{p}_{i+1}$  with balls  $\mathbf{B}_i$ ,  $\mathbf{B}_{i+1}$ , subject to tangent conditions  $\mathbf{t}_i$ ,  $\mathbf{t}_{i+1}$  respectively.

By construction of the problem, the angle  $\alpha_i$  affects only the segment  $\mathbf{S}_i$  as well as the segment  $\mathbf{S}_{i-1}$ , as can be easily seen in Figure 2 (a). Finally, we note that the envelope is fully parameterized by the balls and the spline angles  $\alpha_i$ . Since the balls are fixed, our objective will be to compute the angles  $\alpha_i$  that form the optimal envelope.

### 3 Energy minimization

There are an infinite number of envelopes that are modeled by a contact point on each ball and have a direction tangent to the ball at the point of contact. To further constrain the problem, we require that the envelope have minimal arc length and/or be smooth. We achieve this by finding the angles  $\alpha_i$  that optimize an energy functional. First, we derive equations used to compute the envelope with minimal arc length, then we consider curvature.

#### 3.1 Arc length minimization

In this section, we consider the minimization of arc length. This will result in the shortest envelope that satisfies the geometric constraints imposed by our ball representation. That is, we would like to find the angles  $\alpha_i$  that minimize

$$E_a = \int |\mathbf{S}'| dt, \quad (6)$$

where  $\mathbf{S}'$  is the derivative of  $\mathbf{S}$  with respect to  $t$ . Since the envelope is represented as a set of segments, this is equivalent to

$$E_a = \sum_{i=1}^{N-1} \int |\mathbf{S}'_i| dt, \quad (7)$$

Next, we take the derivative of the energy with respect to the angle  $\alpha_i$ . As stated above, the  $i$ th angle only affects the segments  $\mathbf{S}_{i-1}$  and  $\mathbf{S}_i$ . Therefore,

$$\frac{\partial E_a}{\partial \alpha_i} = \frac{\partial}{\partial \alpha_i} \left( \int |\mathbf{S}'_i| dt \right) + \frac{\partial}{\partial \alpha_i} \left( \int |\mathbf{S}'_{i-1}| dt \right) \quad (8)$$

**First term** Let us consider the first term of Equation 8. Propagating the derivative with respect to  $\alpha_i$  through the integral, it is easy to show that

$$\begin{aligned} \frac{\partial}{\partial \alpha_i} \left( \int |\mathbf{S}'_i| dt \right) &= \int \frac{\partial}{\partial \alpha_i} \langle \mathbf{S}'_i, \mathbf{S}'_i \rangle^{\frac{1}{2}} dt \\ &= \int \langle \mathbf{S}'_i, \mathbf{S}'_i \rangle^{-\frac{1}{2}} \langle \mathbf{S}'_i, \frac{\partial \mathbf{S}'_i}{\partial \alpha_i} \rangle dt, \end{aligned}$$

where  $\langle \rangle$  denotes an inner product. Next, we derive an expression for the  $\frac{\partial \mathbf{S}'_i}{\partial \alpha_i}$  terms using Equation 2, yielding

$$\frac{\partial \mathbf{S}'_i}{\partial \alpha_i} = 3t^2 \frac{\partial \mathbf{A}_i}{\partial \alpha_i} + 2t \frac{\partial \mathbf{B}_i}{\partial \alpha_i} + \frac{\partial \mathbf{C}_i}{\partial \alpha_i} \quad (9)$$

The derivatives  $\frac{\partial \mathbf{A}_i}{\partial \alpha_i}$ ,  $\frac{\partial \mathbf{B}_i}{\partial \alpha_i}$  and  $\frac{\partial \mathbf{C}_i}{\partial \alpha_i}$  can be derived using Equation 3, as

$$\begin{aligned}\frac{\partial \mathbf{A}_i}{\partial \alpha_i} &= 2 \frac{\partial \mathbf{p}_i}{\partial \alpha_i} + \frac{\partial \mathbf{t}_i}{\partial \alpha_i} \\ \frac{\partial \mathbf{B}_i}{\partial \alpha_i} &= -3 \frac{\partial \mathbf{p}_i}{\partial \alpha_i} - 2 \frac{\partial \mathbf{t}_i}{\partial \alpha_i} \\ \frac{\partial \mathbf{C}_i}{\partial \alpha_i} &= \frac{\partial \mathbf{t}_i}{\partial \alpha_i}\end{aligned}\quad (10)$$

Finally, the derivatives  $\frac{\partial \mathbf{p}_i}{\partial \alpha_i}$  and  $\frac{\partial \mathbf{t}_i}{\partial \alpha_i}$  can be derived from Equations 4 and 5 as

$$\begin{aligned}\frac{\partial \mathbf{p}_i}{\partial \alpha_i} &= \begin{bmatrix} -r_i \sin \alpha_i \\ r_i \cos \alpha_i \end{bmatrix} \\ \frac{\partial \mathbf{t}_i}{\partial \alpha_i} &= \begin{bmatrix} -a_i \cos \alpha_i \\ -a_i \sin \alpha_i \end{bmatrix}\end{aligned}\quad (11)$$

We now have all the derivatives needed to compute the first term in Equation 8.

**Second term** Now let us consider the first term of Equation 8, which has a very similar derivation. Propagating the derivative with respect to  $\alpha_i$  through the integral

$$\frac{\partial}{\partial \alpha_i} \left( \int |\mathbf{S}'_{i-1}| dt \right) = \int \langle \mathbf{S}'_{i-1}, \mathbf{S}'_{i-1} \rangle^{-\frac{1}{2}} \langle \mathbf{S}'_{i-1}, \frac{\partial \mathbf{S}'_{i-1}}{\partial \alpha_i} \rangle dt, \quad (12)$$

As before, we derive an expression for the  $\frac{\partial \mathbf{S}'_{i-1}}{\partial \alpha_i}$  terms using Equation 2, yielding

$$\frac{\partial \mathbf{S}'_{i-1}}{\partial \alpha_i} = 3t^2 \frac{\partial \mathbf{A}_{i-1}}{\partial \alpha_i} + 2t \frac{\partial \mathbf{B}_{i-1}}{\partial \alpha_i} + \frac{\partial \mathbf{C}_{i-1}}{\partial \alpha_i} \quad (13)$$

Next, the derivatives  $\frac{\partial \mathbf{A}_{i-1}}{\partial \alpha_i}$ ,  $\frac{\partial \mathbf{B}_{i-1}}{\partial \alpha_i}$  and  $\frac{\partial \mathbf{C}_{i-1}}{\partial \alpha_i}$  can be derived using Equation 3, as

$$\begin{aligned}\frac{\partial \mathbf{A}_{i-1}}{\partial \alpha_i} &= -2 \frac{\partial \mathbf{p}_i}{\partial \alpha_i} + \frac{\partial \mathbf{t}_i}{\partial \alpha_i} \\ \frac{\partial \mathbf{B}_{i-1}}{\partial \alpha_i} &= 3 \frac{\partial \mathbf{p}_i}{\partial \alpha_i} - \frac{\partial \mathbf{t}_i}{\partial \alpha_i} \\ \frac{\partial \mathbf{C}_{i-1}}{\partial \alpha_i} &= \mathbf{0}\end{aligned}\quad (14)$$

We now have all the derivatives needed to compute the second term of Equation 8.

### 3.2 Curvature minimization

We now consider the problem of minimizing curvature. Since curvature can be positive or negative, we choose to minimize the squared curvature, i.e, we would

like to find the angles  $\alpha_i$  that minimize

$$E_c = \int \kappa^2(t) dt, \quad (15)$$

where  $\kappa(t)$  is the curvature of  $\mathbf{S}$  at point  $t$ . Since the envelope is represented as a set of segments, this is equivalent to

$$E_c = \sum_{i=1}^{N-1} \int \kappa_i^2(t) dt, \quad (16)$$

where  $\kappa_i(t)$  is the curvature at point  $t$  along segment  $\mathbf{S}_i$ . Next, we take the derivative of the energy with respect to the angle  $\alpha_i$ . As stated above, the  $i$ th angle only affects the segments  $\mathbf{S}_{i-1}$  and  $\mathbf{S}_i$ . Therefore,

$$\frac{\partial E_c}{\partial \alpha_i} = \frac{\partial}{\partial \alpha_i} \left( \int \kappa_i^2(t) dt \right) + \frac{\partial}{\partial \alpha_i} \left( \int \kappa_{i-1}^2(t) dt \right) \quad (17)$$

Recall that the curvature is given by

$$\kappa_i = \frac{|\mathbf{S}'_i \times \mathbf{S}''_i|}{|\mathbf{S}'_i|^3} = \frac{\langle \mathbf{S}'_i, J\mathbf{S}''_i \rangle}{\langle \mathbf{S}'_i, \mathbf{S}'_i \rangle^{\frac{3}{2}}}, \quad (18)$$

where  $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$  is a 90 degree rotation matrix. Using these equations, Equation 17 becomes

$$\frac{\partial E_c}{\partial \alpha_i} = \frac{\partial}{\partial \alpha_i} \left( \int \left[ \frac{\langle \mathbf{S}'_i, J\mathbf{S}''_i \rangle}{\langle \mathbf{S}'_i, \mathbf{S}'_i \rangle^{\frac{3}{2}}} \right]^2 dt \right) + \frac{\partial}{\partial \alpha_i} \left( \int \left[ \frac{\langle \mathbf{S}'_{i-1}, J\mathbf{S}''_{i-1} \rangle}{\langle \mathbf{S}'_{i-1}, \mathbf{S}'_{i-1} \rangle^{\frac{3}{2}}} \right]^2 dt \right) \quad (19)$$

**First term** Let us consider the second term of Equation 19 first. Propagating the derivative with respect to  $\alpha_i$  through the integral, we see that

$$\begin{aligned} \frac{\partial}{\partial \alpha_i} \left( \int \left[ \frac{\langle \mathbf{S}'_i, J\mathbf{S}''_i \rangle}{\langle \mathbf{S}'_i, \mathbf{S}'_i \rangle^{\frac{3}{2}}} \right]^2 dt \right) = \\ \int 2 \left[ \frac{\langle \mathbf{S}'_i, J\mathbf{S}''_i \rangle}{\langle \mathbf{S}'_i, \mathbf{S}'_i \rangle^{\frac{3}{2}}} \right] \left( \frac{\frac{\partial}{\partial \alpha_i} \langle \mathbf{S}'_i, J\mathbf{S}''_i \rangle}{\langle \mathbf{S}'_i, \mathbf{S}'_i \rangle^{\frac{3}{2}}} - \frac{3 \langle \mathbf{S}'_i, J\mathbf{S}''_i \rangle}{2} \frac{\frac{\partial}{\partial \alpha_i} \langle \mathbf{S}'_i, \mathbf{S}'_i \rangle}{\langle \mathbf{S}'_i, \mathbf{S}'_i \rangle^{\frac{5}{2}}} \right) dt \end{aligned}$$

For this, we need the derivatives  $\frac{\partial}{\partial \alpha_i} \langle \mathbf{S}'_i, J\mathbf{S}''_i \rangle$  and  $\frac{\partial}{\partial \alpha_i} \langle \mathbf{S}'_i, \mathbf{S}'_i \rangle$ . It is easy to show that these derivatives are

$$\begin{aligned} \frac{\partial}{\partial \alpha_i} \langle \mathbf{S}'_i, J\mathbf{S}''_i \rangle &= \langle \frac{\partial \mathbf{S}'_i}{\partial \alpha_i}, J\mathbf{S}''_i \rangle + \langle \mathbf{S}'_i, J \frac{\partial \mathbf{S}''_i}{\partial \alpha_i} \rangle \\ \frac{\partial}{\partial \alpha_i} \langle \mathbf{S}'_i, \mathbf{S}'_i \rangle &= 2 \langle \mathbf{S}'_i, \frac{\partial \mathbf{S}'_i}{\partial \alpha_i} \rangle \end{aligned} \quad (20)$$

Equation 9 gives an expression for  $\frac{\partial \mathbf{S}'_i}{\partial \alpha_i}$ , and from this we see  $\frac{\partial \mathbf{S}''_i}{\partial \alpha_i} = 6t \frac{\partial \mathbf{A}_i}{\partial \alpha_i} + 2 \frac{\partial \mathbf{B}_i}{\partial \alpha_i}$ . The derivatives  $\frac{\partial \mathbf{A}_i}{\partial \alpha_i}$ ,  $\frac{\partial \mathbf{B}_i}{\partial \alpha_i}$  and  $\frac{\partial \mathbf{C}_i}{\partial \alpha_i}$  are given in Equation 10. We now have all the derivatives needed to compute the first term in Equation 19.

**Second term** The first term of Equation 19 is very similar the second term derived above. Propagating the derivative with respect to  $\alpha_i$  through the integral, we see that

$$\begin{aligned} \frac{\partial}{\partial \alpha_i} \left( \int \left[ \frac{\langle \mathbf{S}'_{i-1}, J\mathbf{S}''_{i-1} \rangle}{\langle \mathbf{S}'_{i-1}, \mathbf{S}'_{i-1} \rangle^{\frac{3}{2}}} \right]^2 dt \right) &= \int 2 \left[ \frac{\langle \mathbf{S}'_{i-1}, J\mathbf{S}''_{i-1} \rangle}{\langle \mathbf{S}'_{i-1}, \mathbf{S}'_{i-1} \rangle^{\frac{3}{2}}} \right] \\ &\left( \frac{\frac{\partial}{\partial \alpha_i} \langle \mathbf{S}'_{i-1}, J\mathbf{S}''_{i-1} \rangle}{\langle \mathbf{S}'_{i-1}, \mathbf{S}'_{i-1} \rangle^{\frac{3}{2}}} - \frac{3 \langle \mathbf{S}'_{i-1}, J\mathbf{S}''_{i-1} \rangle \frac{\partial}{\partial \alpha_i} \langle \mathbf{S}'_{i-1}, \mathbf{S}'_{i-1} \rangle}{2 \langle \mathbf{S}'_{i-1}, \mathbf{S}'_{i-1} \rangle^{\frac{5}{2}}} \right) dt \end{aligned} \quad (21)$$

For this, we need the derivatives  $\frac{\partial}{\partial \alpha_i} \langle \mathbf{S}'_{i-1}, J\mathbf{S}''_{i-1} \rangle$  and  $\frac{\partial}{\partial \alpha_i} \langle \mathbf{S}'_{i-1}, \mathbf{S}'_{i-1} \rangle$ . It is well known that the derivative of the cross product is

$$\begin{aligned} \frac{\partial}{\partial \alpha_i} \langle \mathbf{S}'_{i-1}, J\mathbf{S}''_{i-1} \rangle &= \langle \frac{\partial \mathbf{S}'_{i-1}}{\partial \alpha_i}, J\mathbf{S}''_{i-1} \rangle + \langle \mathbf{S}'_{i-1}, J \frac{\partial \mathbf{S}''_{i-1}}{\partial \alpha_i} \rangle \\ \frac{\partial}{\partial \alpha_i} \langle \mathbf{S}'_{i-1}, \mathbf{S}'_{i-1} \rangle &= 2 \langle \mathbf{S}'_{i-1}, \frac{\partial \mathbf{S}'_{i-1}}{\partial \alpha_i} \rangle \end{aligned} \quad (22)$$

Equation 13 gives an expression for  $\frac{\partial \mathbf{S}'_{i-1}}{\partial \alpha_i}$ , and from this we see  $\frac{\partial \mathbf{S}''_{i-1}}{\partial \alpha_i} = 6t \frac{\partial \mathbf{A}_{i-1}}{\partial \alpha_i} + 2 \frac{\partial \mathbf{B}_{i-1}}{\partial \alpha_i}$ . The derivatives  $\frac{\partial \mathbf{A}_{i-1}}{\partial \alpha_i}$ ,  $\frac{\partial \mathbf{B}_{i-1}}{\partial \alpha_i}$  and  $\frac{\partial \mathbf{C}_{i-1}}{\partial \alpha_i}$  are given in Equation 14. Thus, all the derivatives needed to compute the second term of Equation 19 have been derived.

In these equations, we evaluate the integrals in Equations 8 and 19 for each angle  $\alpha_i$ . However, for the first ball,  $i = 1$ , there is no segment  $\mathbf{S}_{i-1}$ , so we ignore the integral this term. Likewise, for the last ball,  $i = N$ , there is no segment  $\mathbf{S}_i$ , so we ignore the integral this term.

### 3.3 Discussion

To summarize, we have derived the gradient of energy functionals  $E_a$  and  $E_c$  with respect to angles,  $\alpha_i$ . The derivation inherently consisted of several steps via the chain rule, as the energy is the a combination of the arc length and squared curvature, which in turn are a functions of the envelope, which in turn is a function of the segment constants  $\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i, \mathbf{D}_i$  and  $\mathbf{A}_{i-1}, \mathbf{B}_{i-1}, \mathbf{C}_{i-1}, \mathbf{D}_{i-1}$ , which in turn are functions of the angles  $\alpha_i$ .

### 3.4 Implementation

We combine the energies  $E_a$  and  $E_c$  together, as

$$E = (1 - k)E_a + kE_c \quad (23)$$



where  $k$  is a constant used to weight the arc length minimization relative to the curvature minimization. Convex combinations of the two can be selected using  $k \in [0, 1]$ . Therefore, the combined energy minimization is given by

$$\frac{\partial E}{\partial \alpha_i} = (1 - k) \frac{\partial E_a}{\partial \alpha_i} + k \frac{\partial E_c}{\partial \alpha_i} \quad (24)$$

where  $\frac{\partial E_a}{\partial \alpha_i}$  is given in Equation 8 and  $\frac{\partial E_c}{\partial \alpha_i}$  is provided in Equation 17. In all of the experiments in this paper, we fix  $k = 0.9$ , to encourage smoother solutions.

These equations are a set of differential equations that can be used in a gradient descent procedure to optimize the envelope by manipulating the angles  $\alpha = [\alpha_1, \dots, \alpha_N]^T$ . Let  $\alpha_i^n$  be the  $i$ th angle at iteration  $n$ . We can then update the angles by moving them in the negative gradient direction, i.e.,

$$\alpha^{n+1} = \alpha^n - \Delta t \nabla E_{\alpha^n}, \quad (25)$$

where  $\Delta t$  is a time step and  $\nabla E_{\alpha^n} = [\frac{\partial E}{\partial \alpha_1^n}, \frac{\partial E}{\partial \alpha_2^n}, \dots, \frac{\partial E}{\partial \alpha_N^n}]^T$ .

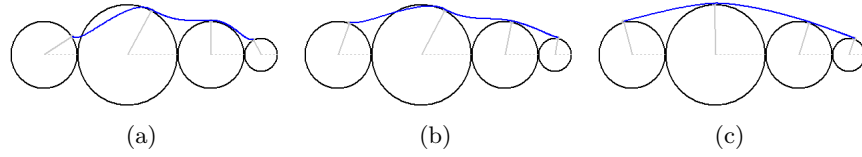
The computational complexity of the algorithm depends on the number of balls  $N$  and the number of points  $L$  on a segment where the points and derivatives are evaluated. For each iteration of the gradient descent procedure, the computational complexity is  $O(NL)$ . The number of iterations required depends on the time step  $\Delta t$  as well as how close the initial envelope is to the final solution.

## 4 Results

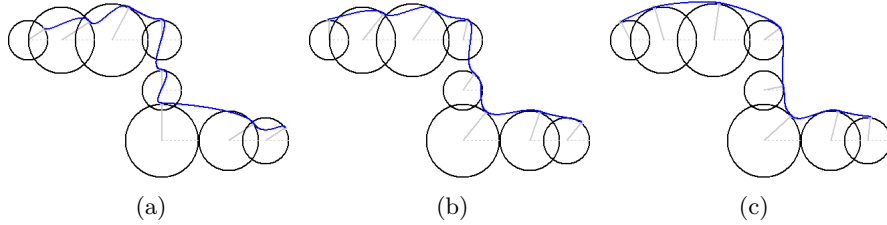
A simple example is provided in Figure 3. Here, four balls of radius 50, 75, 50, and 25 pixels, respectively were set along the  $x$ -axis. The initial angles for this experiment were 0.57, 1.07, 1.57 and 2.07 radians, respectively; the initial envelope is shown in part (a) of the figure. The angles were iteratively updated using Equation 25. An intermediate solution after 50 iterations is shown in (b), at this stage, the envelope is considerably smoother while still satisfying the constraints of the problem. We show the result after 100 iterations in (c), at which point the energy has reached a minimum and the angles have converged. The solution (all 100 iterations) is computed in 47 milliseconds using C++ code compiled on a machine with a 3.0 GHz single-core processor.

Figure 4 shows a slightly more complicated example for which some balls overlap and others do not. The initial envelope is shown in (a), an intermediate result after 70 iterations in (b), and the final result upon convergence after 140 iterations in (c). The solution (all 140 iterations) is computed in 143 milliseconds.

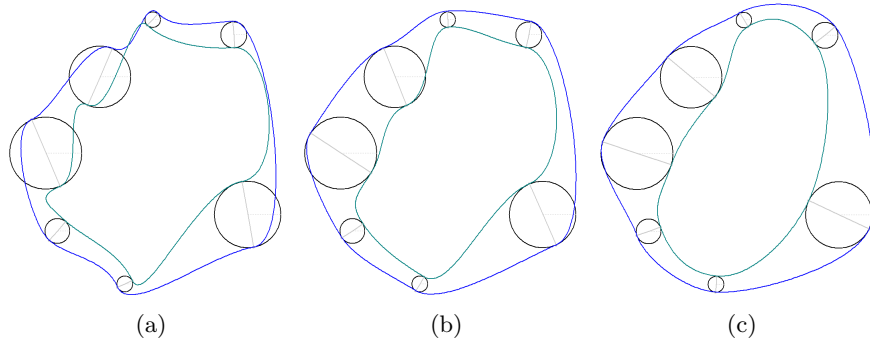
In Figure 5 we show an example of generating an interpolating region for a collection of balls. In this case, we have two envelopes, one defining the interior boundary of the region (rendered in green), and another defining the exterior boundary (rendered in blue). For each ball, there are two points of contact: one from the interior envelope and one for the exterior envelope; however, we constrain these points of contact to be separated by 180 degrees. Therefore, for each ball there is only one angle  $\alpha_i$  to be determined as in the examples above.



**Fig. 3.** Example ball skinning. The initialization is shown in (a), and the result after 50 iterations is shown in (b), and the converged result after 100 iterations is shown in (c). The envelope is rendered in a blue color.



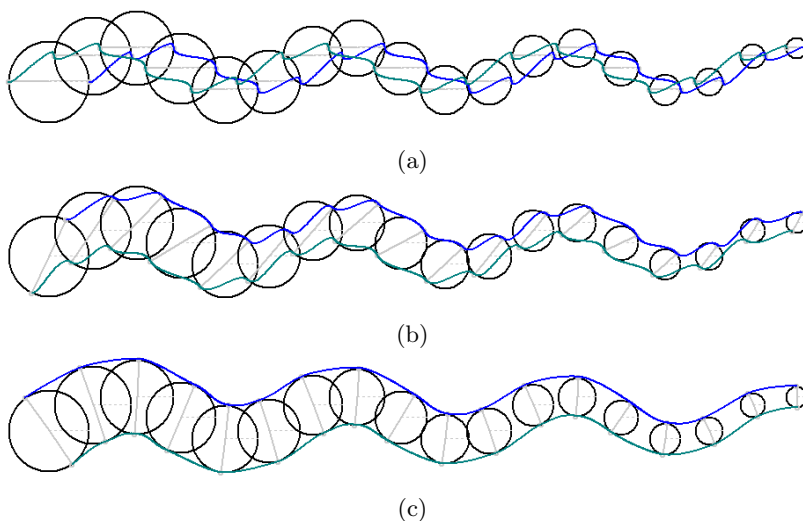
**Fig. 4.** Another skinning of a set of balls. Initialization (a), intermediate result (b) after 70 iterations, and final result upon convergence (c) after 140 iterations.



**Fig. 5.** Generating a smooth interpolating region between a set of balls. Initialization (a), intermediate result (b) after 50 iterations, and final result upon convergence (c) after 100 iterations.

We solve for the angle for all the balls, with each envelope contributing a term in Equation 25. In part (a) of Figure 5 we show an initialization, in (b) and intermediate result after 50 iterations, and in (c) the final converged result after 100 iterations. Convergence for this example occurs in 190 milliseconds.

More examples are provided in Figures 6 and 1. In Figure 6, the balls are arranged on a sine wave and have a variable radius. Convergence of the skinning algorithm, starting from a set of angles far from the optimal result, takes 775



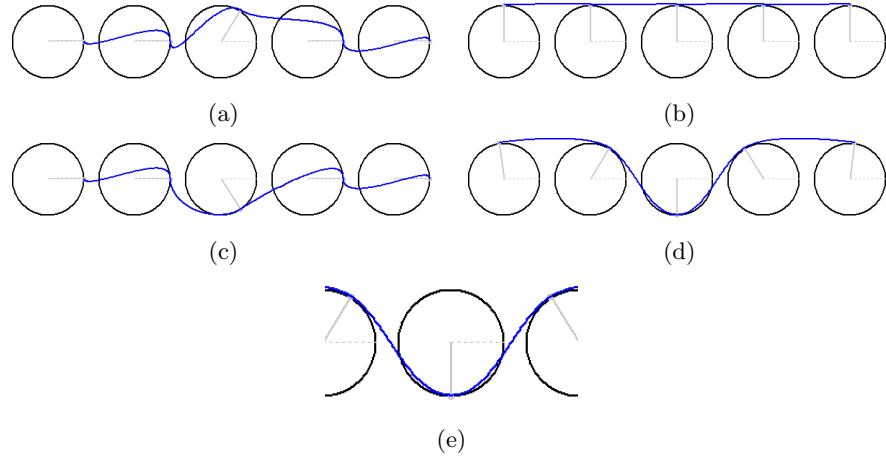
**Fig. 6.** Generating a smooth interpolating region between a set of balls. Initialization (a), intermediate result (b) after 80 iterations, and final result upon convergence (c) after 160 iterations.

milliseconds. In Figure 1, the variable radius balls are arranged in a spiral. The envelope is generated in 2.5 seconds.

We note that our gradient descent approach only guarantees a locally optimal solution; the particular solution will depend on the convexity of the energy functional as well as the initial condition, as demonstrated in Figure 7. In (a) we show an initial envelope, and in (b) the result of our approach. The initial condition in (c) is identical to (a) except the middle ball has a different angle drawing the envelope down. In (d), we show the result of our approach starting from (c), resulting in a different solution. In this example and many others in the paper, the initial envelopes are chosen to be far from the final solution to demonstrate the effect of the differential equations. However, in practice, it is typically easy to determine a good initialization based on choosing an angle for each ball that is along the ray orthogonal to the centerline connecting adjacent ball centroids. Finally, we note that the envelope our method generates may pass through a ball (shown in Figure 7 (e)) since it is only constrained to be tangent to the ball at one point of incidence. In our application of modeling blood vessels, this is an acceptable solution since the ball itself is a geometric proxy of the local vessel geometry.

## 5 Conclusion

In this paper, we presented a method for optimally skinning an ordered set of 2D balls. Our formulation of the problem requires that the envelope be modeled



**Fig. 7.** Solution depends on initial condition. Different solutions (b) and (d) are possible depending on the initial condition (a) and (c), respectively. In (e), we show that the envelope may pass through a ball.

by a point of contact with each ball and at the point of contact, be tangent to the ball. We have presented novel derivations resulting in differential equations that minimize a convex combination of the arc length and curvature of a third order polynomial spline subject to these constraints. Starting with an initial envelope, we evolve the envelope’s parameters until convergence. Experimental results demonstrate the viability of the method.

For future work, we are interested in extending the approach to interpolate balls in  $R^3$ . In this case, the point of contact for a ball will be modeled with two angles, and the derivation will result in a coupled set of partial differential equations for these two angles. The envelope will then be a surface that interpolates the balls, constrained by the points of contact.

## References

1. K. Singh and E. Kokkevis, “Skinning Characters using Surface Oriented Free-Form Deformations,” in *Graphics Interface*, 2000, pp. 35–42.
2. H. Cheng and X. Shi, “Quality Mesh Generation for Molecular Skin Surfaces Using Restricted Union of Balls,” in *IEEE Visualization*, 2005.
3. H. Edelsbrunner, “Deformable smooth surface design,” *Discrete and Computational Geometry*, vol. 21, no. 1, pp. 87–115, 1999.
4. N. Kruithov and G. Vegter, “Envelope Surfaces,” in *Annual Symposium on Computational Geometry*, 2006, pp. 411 – 420.
5. B. Whited, J. Rossignac, G. Slabaugh, T. Fang, and G. Unal, “Pearling: 3D Interactive Extraction of Tubular Structures from Volumetric Images,” in *Interaction in Medical Image Analysis and Visualization, held in conjunction with MICCAI*, 2007.