The Visual Computer

Variable-radius blending by constrained spine generation

Jung-Hong Chuang,
Wei-Chung Hwang

Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China

Radius blends, very important in geometric and solid modeling, can be seen as the trimmed envelope of a rolling sphere or a sweeping circle with a constant or variable radius that centers on a spine curve and touches the surfaces to be blended along the linkage curves. Usually, in variable-radius blending, the radius is difficult to specify, and the spine curve is hard to trace. We propose several geometric constraints to specify the variable radius, which we then translate to a nonlinear system to represent the spine curve exactly. This is finally traced numerically in a high-dimensional space. We also propose a paradigm that implements the constraints while tracing along the spine curve in 3D space. We represent the result in parametric form.

Key words: Geometric modeling – Blending – Variable-radius spherical and circular blend – Spine specification

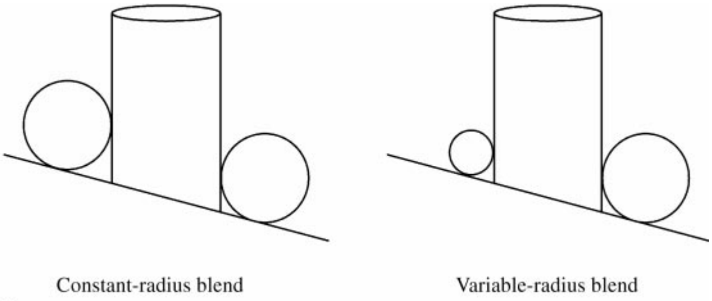
Correspondence to: J.-H. Chuang

1 Introduction

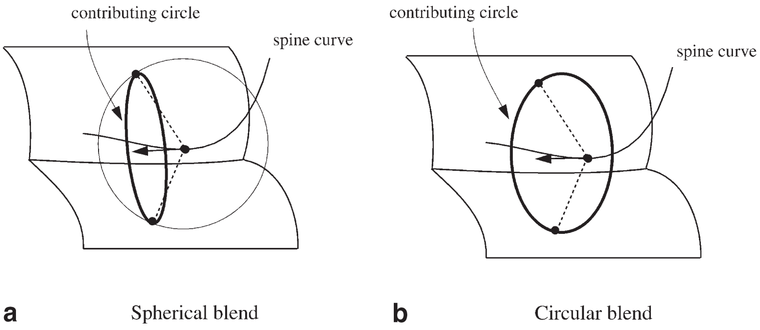
Surface blending is becoming a necessary operation in geometric and solid modeling in which sharp edges must often be rounded or replaced by a smooth surface for aesthetic or functional reasons. A blending surface is a surface that smoothly connects two or more surfaces, which are called *base surfaces*. The curves where the blend meets the base surfaces are called *linkage curves*, *rail curves*, or *contact curves*. The base surfaces can be either intersecting or disjointed. When the base surfaces are disjointed, the blending surface smoothly joins them. While the shape of blends is not critical in some applications, positional and tangential continuity between the blend and base surfaces is generally required along the linkage curves.

The base surfaces can be defined either implicitly or parametrically. The blending of implicit surfaces has been treated extensively in the literature (Hoffmann and Hopcroft 1987; Middleditch and Sears 1985; Rockwood and Owen 1987). These blending techniques result in an exact implicit blend, frequently in a rigorous mathematical form. Most existing methods for blending parametric surfaces derive the blending surface by requiring positional and tangential continuity along the linkage curve on each of the base surfaces (Filip 1989; Koparkar 1991) and produce blends in the form of approximated spline surfaces. Methods for implicit blends generally cannot be extended to parametric surfaces unless a complex implicitization process is applied. However, several recent efforts have been made to extend implicit methods to the blending of parametric surfaces (Chuang and Lien 1995; Lee et al. 1993; Vaishnav and Rockwood 1993).

Among the blending methods currently available, radius blends are popular because of their simple geometric description. Radius blends are generally described as the trimmed envelope of a rolling sphere or a sweeping circle. Pegna and Wilde (1990) term the first of these descriptions a *spherical blend* and the second, a *circular blend*. The trajectory of the center of the rolling sphere or sweeping circle is usually called the *spine curve*. The radius of the rolling sphere and the sweeping circle can be constant or can vary along the spine curve. A *constant-radius blend* requires the radius value as input and can generally be derived automatically by the modeler; a *variable-radius blend* generally requires more intervention from the user. Although a constant-radius blend is geometri-



1



2

Fig. 1. Radius blends of a cylinder and an inclined plane

Fig. 2a–b. Point correspondence and contributing circle for the variable-radius blend; **a** spherical blend; **b** circular blend

cally simpler and computationally more efficient than a variable-radius blend, constant-radius blends are inadequate in certain applications. For example, when a cylinder is blended with an inclined plane, as shown in Fig. 1, a constant-radius blend results in a blend that has a circular arc with a radius larger than necessary for a certain portion of the blend. In variable-radius blends, it is, however, difficult to define the spine curve and the variable radius and to form a correct point correspondence between points on the spine curve and the respective linkage curves. Previously proposed methods either explicitly specify the radius as a radius function defined on a given reference curve (Chuang et al. 1995; Harada et al. 1991) or implicitly define the spine curve as the intersection of a given reference surface with the Voronoi or equidistant surface of two base surfaces (Chandru et al. 1990; Hoffmann 1990). Such specifications for a suitable radius function usually require experience and lots of work.

In this paper, we propose several constraints that can be used to specify the variable radius and can easily be adopted in the formulation of spine curves and variable-radius blendings. Together with the higher-dimensional representation of Voronoi surfaces (Hoffmann 1990), we formulate the radius constraint in a nonlinear system that exactly represents the spine curve. Alternatively, the constraints can be implemented in a marching procedure that marches along the spine curve in R^3. Although tracing the spine curve in higher-dimensional space using the exact representation of the spine is conceptually simpler, it is computationally more expensive than tracing in R^3. In Sect. 2, we review methods for specifying and manipulating radius blends. In Sect. 3, we present several possible constraints and a general constraint for specifying variable-radius. In Sect. 4, an exact higher-dimensional formulation for the variable-radius spine curve that incorporates the proposed constraints is proposed and the tracing of the spine curve is described. In Sect. 5, we propose a procedure that implements the constraints while tracing the spine curve in R^3. Surface formulations for the spherical and circular blends are described in Sect. 6. In Sect. 7, we show some experimental results and compare the proposed 3D tracing to high-dimensional tracing. Sect. 8 presents concluding remarks.

2 Review of radius blends

In the literature, spherical blends are represented either exactly or approximately, while circular blends are usually approximated. The constant- or variable-radius spherical blend for base surfaces in either implicit or parametric form can be formulated exactly in higher-dimensional spaces with the *dimensionality paradigm* (Hoffmann 1990). The idea is to represent the blend as the natural projection of a 2-surface in n-space into 3-space, for some $n > 3$. For example, the spine of the constant-radius blend with radius r is the intersection of offsets with radius r from each base surface. The offset of base surfaces is represented as the envelope of spheres with radius r and center on the respective surface. The spine curve of the variable-radius blend can be defined as the intersection of a third user-defined surface, called the *reference surface*, with the surface equidistant from two base surfaces, called the Voronoi (or equidistant) surface of the two base surfaces. The Voronoi surface can again be defined as the intersection of offsets with varying radius from each base surface in which the radius is itself a variable. The constant- or variable-radius spherical blend is then simply the envelope of spheres centering on the derived spine curve and having a suitable radius that is specified either explicitly as a constant r or implicitly as the distance from the sphere center to its foot points on the base surfaces. Such an envelope is the solution set of a system of equations each of which is converted from some geometric constraints. Each solution point of the system represents a spine point, parameter values for foot points of the spine point on each of the base surfaces, and other auxiliary parameters. According to the constraints, points on two linkage curves correspond to each other if the surface normals at these two points are coplanar and intersect at the corresponding spine point. This type of point correspondence for constant-radius spherical blends can easily be found by tracing the spine curve, which is often defined as the intersection of offsets with the specified radius from the two base surfaces. The derivation of point correspondence on the spine and linkage curves in variable-radius spherical blends is much more complicated, however, since the spine curve cannot easily be defined. It is worthing noting that the contributing circle for the constant-radius spherical and circular blend is the greatest circle passing through the sphere center and its foot points. For the variable-radius spherical blend, the contributing circle is on the plane containing the two foot points of the sphere center and perpendicular to the tangent of the spine curve at the center of the sphere (see Fig. 2a).

Other proposed methods approximate radius blends using parametric surfaces. Methods for approximating constant-radius spherical blends are presented in (Barnhill et al. 1993; Choi and Ju 1989; Klass and Kuhn 1992; Varady 1989). All of these methods derive the spine curve by tracing the intersection of offsets with an equal radius from the base surfaces. For variable-radius spherical blends, a marching procedure is proposed by Chuang et al. (1995) that ensures the correct point correspondence while tracing the spine and linkage curves simultaneously in R^3.

Harada et al. (1991) propose a *sliding-circle method* to derive the spine curve for the variable-radius circular blend. In this method, the user defines a control curve and a radius function on it. A circle is then slid onto two parametric base surfaces so that it is on the normal plane of the control curve and its radius is the value of the radius function at the respective point on the control curve. Choi (1991) assumes the radius function to be monotonic between a minimum radius and a maximum radius, both of which are chosen by the user. The intersection curve of two offsets with a radius equal to the average of the minimum and maximum radius values is traced and used as a control curve. Then a circle with a suitable radius is derived on the plane normal to the constant-radius spine. For these two approaches, points on linkage curves correspond to each other when they are on the normal plane of a user-defined control curve, and normals of two corresponding linkage points are not necessarily coplanar. Thus, the correspondence is critically dependent on the control curve. Moreover, the derived sweeping circle is centered on the spine curve and lies on the normal plane of the control curve. Note that the sweeping circle for the variable-radius circular blends implemented in this paper is the greatest circle of the rolling sphere that passes foot points of the sphere center (Fig. 2b).

3 Constraints for the variable-radius function

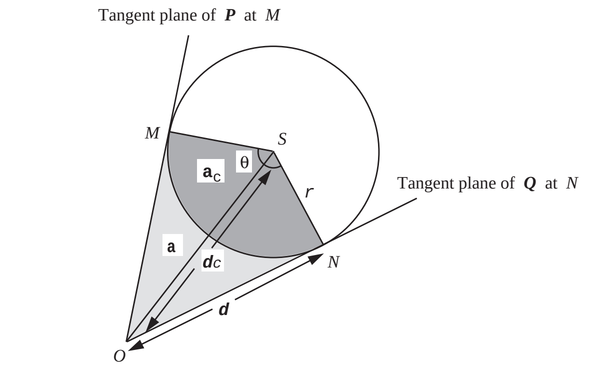

Tangent plane of $\boldsymbol{P}$ at M
M
S
$\mathbf{a_c}$
θ
r
Tangent plane of $\boldsymbol{Q}$ at N
$\mathbf{a}$
$\mathbf{dc}$
N
$\mathbf{d}$
O

Fig. 3. Specifying the variable-radius function

Current geometric modelers usually require a user-prescribed radius function or reference surface for generating variable-radius blends. Since the geometry along the intersection of two base surfaces can be complicated, specifying an appropriate radius function is a nontrivial task. This type of scheme greatly limits the design capability, and for this reason, most modelers allow only linear radius functions. Here we present several constraints from which the variable radius can automatically be generated.

Let P and Q be two base surfaces and S be a point on the spine curve. Then we have the radius function

$$r = d(S, P) = d(S, Q),$$

where $d(S,P)$ and $d(S,Q)$ are the orthogonal distances from S to P and Q, respectively. With only one user-defined parameter, the proposed constraints can specify the varying radius according to the angle between line segments from the center of the rolling sphere to two contact points on base surfaces. Figure 3 depicts this configuration, in which r and S are the radius and the center of the rolling sphere, and M and N are contact points of the rolling sphere on base surfaces P and Q, respectively. O is the intersection of the tangent plane of P and Q at M and N, respectively, and the plane determined by S, M, and N, a represents the area inside the triangle formed by M, O, and N and outside the circular arc from M to N, a_c represents the circular area from M to N, θ represents the angle between line segments SM and SN, d is the distance from O to N, and d_c is the distance from O to S. Note that, since M and N are contact points of the rolling sphere on base surfaces P and Q, respectively, vectors from M to S and from N to S are normals to the surface P and Q, respectively. Some possible constraints for the variable radius are:

- *Constant radius.* The radius r is set to be constant:

  $$r = R.$$

- *Constant arc length.* The length l of the circular arc from M to N is set to be a constant L. Since $l = r\theta$, we have

  $$r = \frac{L}{\theta}.$$

- *Constant circular area.* The area a_c is set to be a constant A_c. Since $a_c = r^2 \frac{\theta}{2}$, we have

  $$r = \sqrt{\frac{2A_c}{\theta}}.$$

- *Constant area.* The area a is set to be a constant A. Since $a = r^2 \tan\frac{\theta}{2} - r^2 \frac{\theta}{2}$, we have

  $$r = \sqrt{\frac{A}{\tan\frac{\theta}{2} - \frac{\theta}{2}}}.$$

- *Constant range distance.* The distance d is set to be a constant D. Since $d = r\tan\frac{\theta}{2}$, we have

  $$r = \frac{D}{\tan\frac{\theta}{2}}.$$

- *Constant circular distance.* The distance d_c is set to be a constant D_c. Since $d_c = \frac{r}{\cos\frac{\theta}{2}}$, we have

  $$r = D_c \cos\frac{\theta}{2}.$$

All these constraints can be generalized as

$$r(\theta) = \frac{c}{f(\theta)}, \tag{1}$$

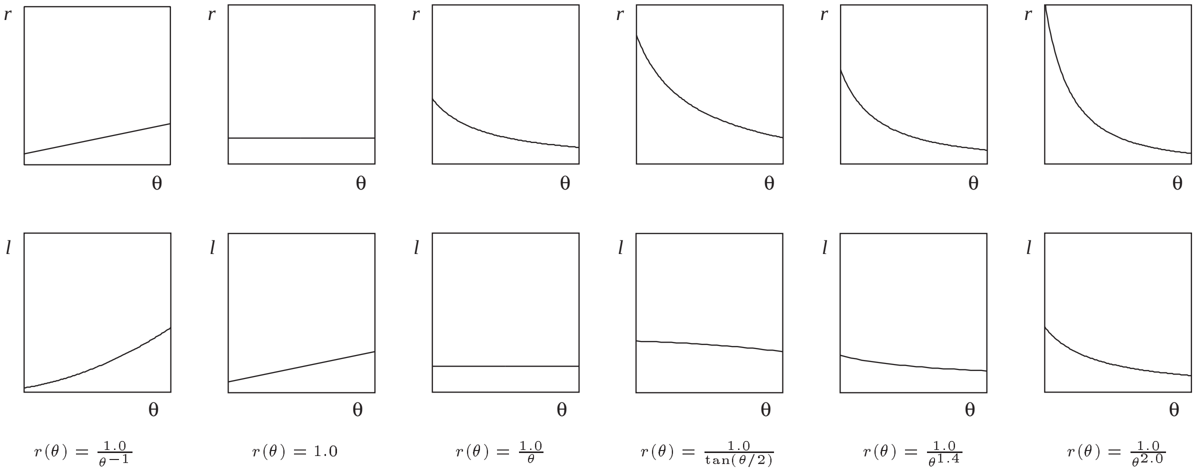


Fig. 4. Top: relation between the radius and θ, bottom: relation between the circular arc length and θ

where c is a user-specified constant and $f(\theta)$ is a non-negative function of θ, $0 < \theta < \pi$. The function $f(\theta)$ basically controls how the radius varies with respect to the angle θ. For example, if $f(\theta)$ is monotonely increasing (or decreasing), $r(\theta)$ becomes smaller (or larger) as θ increases. Thus, an appropriate $f(\theta)$ can easily be chosen according to the particular design needs. Figure 4 depicts the relation between the radius and θ (top) as well as the relation between the length of circular arc and θ (bottom) for some pairs of c and $f(\theta)$, where $0.4 \leq \theta \leq \pi/2$. To derive a variable-radius blend, users are only required to set a function $f(\theta)$ and the value of c. Because they allow an appropriate radius to be specified and require only a very simple user input, the proposed constraints should greatly facilitate the derivation of variable-radius blends.

4 An exact variable-radius spine and its tracing

The radius function described in Eq. 1 can easily be augmented to the high-dimensional formulation of a Voronoi surface proposed by Hoffmann (1990). It does not matter if the base surfaces are implicitly or parametrically defined. In this section, we discuss the case of parametric base surfaces. Given two parametric surfaces

$$P(s,t) = \big(P_x(s,t), P_y(s,t), P_z(s,t)\big)$$

and

$$Q(u,v) = \big(Q_x(u,v), Q_y(u,v), Q_z(u,v)\big),$$

the Voronoi surface of P and Q can be defined as the intersection of r-offsets of P and Q by the following system of six equations in eight variables x, y, z, s, t, u, v, and r:

$$S_P : (x - P_x(s,t))^2 + \big(y - P_y(s,t)\big)^2 + (z - P_z(s,t))^2 - r^2 = 0 \qquad (2)$$

$$\frac{\partial S_P}{\partial s} = 0 \qquad (3)$$

$$\frac{\partial S_P}{\partial t} = 0 \qquad (4)$$

$$S_Q : (x - Q_x(u,v))^2 + \big(y - Q_y(u,v)\big)^2 + (z - Q_z(u,v))^2 - r^2 = 0 \qquad (5)$$

$$\frac{\partial S_Q}{\partial u} = 0 \qquad (6)$$

$$\frac{\partial S_Q}{\partial v} = 0. \qquad (7)$$

Equations 2–4 and Eqs. 5–7 represent the offsets of P and Q, respectively, with an unknown radius r. Since r itself is a variable, the intersection of r-offsets of P and Q represents points equidistant to P and Q. Figure 5 illustrates the Voronoi surface of a cylinder and an inclined plane. Note that Eq. 2 represents the sphere S_P centered on the surface point $P(s,t)$ with radius r, while Eq. 3 and 4 constrain points on the sphere S_P to the offset of P.

The natural projection of the zero set of the system into (x,y,z)-space represents the desired Voronoi surface. Since each point on the Voronoi surface is equidistant to P and Q with a variable-radius r, we can specify a spine curve by requiring that r satisfy

$$rf(\theta) - c = 0 \tag{8}$$

for some user-defined c and $f(\theta)$, where θ is the angle between two unit surface normals at $P(s,t)$ and $Q(u,v)$. That is, $\theta = \arccos(\mathbf{m}\cdot\mathbf{n})$, where $\mathbf{m}$ and $\mathbf{n}$ are unit surface normals at $P(s,t)$ and $Q(u,v)$, respectively. The expression for θ can be translated to an equation such as

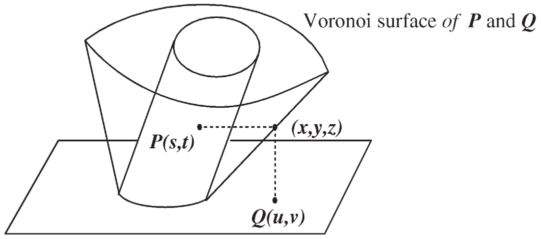
$$\theta - \arccos \left[\frac{[x-P_x(s,t), y-P_y(s,t), z-P_z(s,t)]}{\|[x-P_x(s,t), y-P_y(s,t), z-P_z(s,t),]\|} \cdot \frac{[x-Q_x(u,v), y-Q_y(u,v), z-Q_z(u,v)]}{\|[x-Q_x(u,v), y-Q_y(u,v), z-Q_z(u,v)]\|} \right] = 0. \tag{9}$$

Appending Eqs. 8 and 9 to Eqs. 2–7 yields a system of equations that provides an exact formulation for the desired variable-radius spine curve.

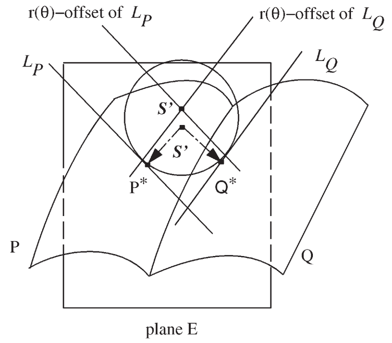
The spine curve so defined can be traced with the algorithm described by Hoffmann (1990). Although such a tracing scheme is simple, the computational cost involved for computing the local approximant and performing point refinement with a Newton iteration increases with the number of dimensions. Evaluating surface partials becomes very complicated, especially for NURBS surfaces. The derivation of θ in Eq. 9 requires partials that are one degree higher than those for the local approximant, as shown in Eq. 9, and this further complicates the computation. Moreover, it becomes nontrivial in R^9 to determine the step size for tracing, especially near the boundaries of surfaces, and to decide appropriate tolerances for point refinement. This high-dimensional tracing is harder to implement and computationally much more expensive than the proposed tracing in R^3. See Section 7 for a comparison.

5 Tracing the spine and linkage curves in R^3

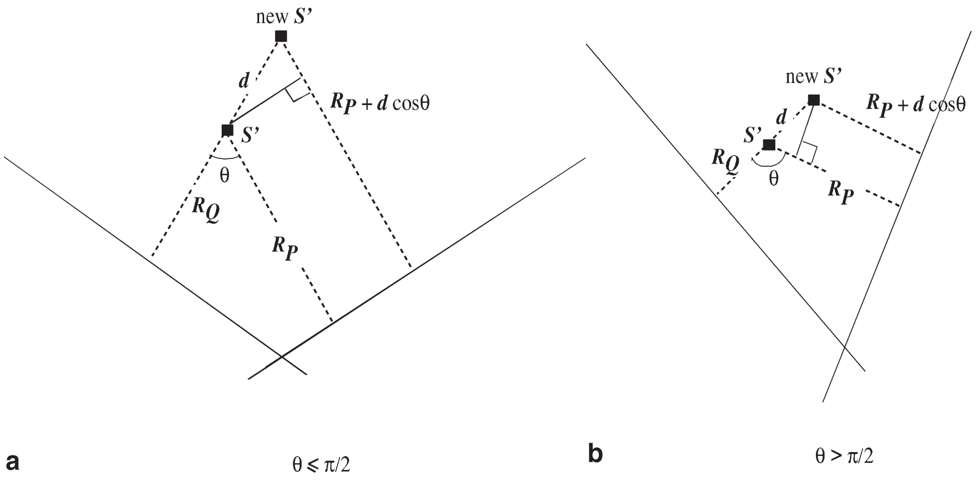
The spine for the constant-radius blend can be represented as the intersection of offsets of two base surfaces. When the offset is represented in differential form, the constant-radius spine can be traced in R^4 with a system of three nonlinear equations for the spine. However, the point refinement requires second derivatives of the base surfaces and solving a 3×4 linear system for each iterate. When the offset is represented by the dimensional paradigm, the constant-radius spine can be traced in R^7 with a system of six nonlinear equations for the spine. The point refinement requires solving a 6×7 linear system for each iterate. Hence, tracing the constant-radius spine in either R^4 or R^7 is conceptually simple, but the computational cost can be expected to be high. Tracing the spine in R^3 and performing point refinement without the use of the nonlinear system for the spine is generally considered more efficient (Barnhill et al. 1993; Chuang et al. 1995; Varady 1989). Tracing the variable-radius spine in R^9 with the exact representation raises similar problems. Thus, tracing the variable-radius spine in R^3 is generally preferable; however, the tracing direction cannot be determined as easily as that for the constant-radius spine, and point refinement is much more complicated, because the radius has to be determined simultaneously in the refinement process; see (Chuang et al. 1995). For a given pair of parametric surfaces $P(s,t)$ and $Q(u,v)$ to be blended and a constraint for radius function as in Eq. 1, we present a marching procedure that starts from a point on the spine curve, steps along the tangent vector of the spine curve, and refines the next approximate point to a spine point that satisfies the given radius constraint. The refinement is performed in R^3. The result of the marching procedure is a set of triples, each



5



6



7

Fig. 5. Voronoi surface of a cylinder and an inclined plane

Fig. 6. Toward a new spine point in step 2

Fig. 7a, b. The point displacement in step 1

consisting of a point on the spine curve and its foot points on P and Q. The marching procedure is outlined as follows:

1. If the starting point is not on the spine curve, refine it to a spine point S that satisfies the constraint given for the radius.
2. Derive the step vector *step_vector* in the tangential direction of the spine curve at S and obtain the next approximate point S'.
3. Refine S' to a spine point S that satisfies the constraint given for the radius.
4. Repeat steps 2 and 3 until one of the foot points of S is outside the domain of the respective base surface or S is equal to (within a distance tolerance) the starting spine point, which indicates a closed spine.

If the spine is an open curve, the procedure must be called again with the opposite tracing direction, and two segments should be merged to form the complete open spine.

Locating starting points on the spine is in general difficult. An approach that works well in practice is first to locate an intersection point between

$P(s,t)$ and $Q(u,v)$ by surface subdivision (Barnhill and Kersey 1990) and then to refine the point to the spine with a refinement procedure, which is described in Sect. 5.2.

The derivation of the step vector is described in Sect. 5.1, and the point refinement method is presented in Sect. 5.2.

5.1 Step vector

The step vector considered here is in the tangential direction of the variable-radius spine curve. The tangent of the constant-radius spine at a point is simply in the direction of the cross-product of surface normals at foot points of the spine point. For a variable-radius spine, the exact tangent of the spine cannot be obtained from such a simple paradigm; we need the exact representation of the variable-radius spine. To obtain this, we first compute the tangent vector, say $\mathbf{t}$, of the spine curve in R^9 by requiring that $\mathbf{t}$ be perpendicular to the gradient of each function in Eq. 2–9. This amounts to solving a system of eight linear equations in nine variables. The tangent vector of the spine curve is then the natural projection of $\mathbf{t}$ into (x,y,z)-space (Chuang and Hoffmann 1992).

The step size can be fixed or variable. A fixed step size tends to be conservative and slows down the tracing. The variable step size can be estimated from the local curvature which, however, has to be done beforehand. An alternative, proposed by Barnhill et al. (1993), is to determine the step size according to the angle between the unit tangent vectors at two successive spine points.

5.2 Point refinement

Since the spine is not represented explicitly, the refinement process has to bring the approximated spine point S' to a point that satisfies the radius constraint. Each refining iteration consists of three steps. First, we refine S' to a point that is equidistant to two base surfaces. Second, we derive a radius by applying the radius constraint to the resulting configuration. Finally, if the orthogonal distance from S' to the base surfaces is equal to the derived radius, we have found the point; otherwise, we move S' to a point that is more likely to have the derived radius as its orthogonal distance to the base surfaces and denote this point by S'. The procedure is as follows:

1. Refine S' to a point that is equidistant to the base surfaces:

1a. Find the projection points P^* and Q^* of S' onto $P(s,t)$ and $Q(u,v)$, respectively.
Let E be the plane passing through S', P^*, and Q^*.

1b. Let $R_P = \|S' - P^*\|$ and $R_Q = \|S' - Q^*\|$.
If $|R_P - R_Q| < \mathit{dist\_tolerance}$, then go to Step 2.

1c. Compute unit normals N_P and N_Q of $P(s,t)$ and $Q(u,v)$ at P^* and Q^*, respectively.
Compute the angle θ between N_P and N_Q.
If $R_P > R_Q$
then set S' as $S' + \frac{R_P - R_Q}{1-\cos\theta} N_Q$ and compute the projection P^* of S' onto P,
else set S' as $S' + \frac{R_Q - R_P}{1-\cos\theta} N_P$ and compute the projection Q^* of S' onto Q.

1d. Let E be the plane passing through S', P^*, and Q^*.
Go to Step 1b.

2. Compute the constrained radius and bring S' to a point that is a better approximation having the derived radius as its orthogonal distance to the base surfaces:

2a. Derive the radius $r(\theta)$ according to the radius constraint.

2b. If $|R_P - r(\theta)| < \mathit{dist\_tolerance}$ and $|R_Q - r(\theta)| < \mathit{dist\_tolerance}$, then the refinement succeeds and stops.

2c. Calculate the intersection line L_P between E and the tangent plane of $P(s,t)$ at P^*, and the intersection line L_Q between E and the tangent plane of $Q(u,v)$ at Q^* (Fig. 6).

2d. Find a circle with radius $r(\theta)$ that lies on E and touches both L_P and L_Q. Set the new approximate spine point S' as the center of the circle (Fig. 6).

2e. Go to Step 1.

Step 1 and the whole point refinement process may either succeed in finding the designated point or fail to do so within a user-specified maximum number of iterations. When the refinement fails, the current step size should be reduced and the point refinement repeated.

The displacement from S' to $S' + \frac{R_P - R_Q}{1-\cos\theta} N_Q$ or $S' + \frac{R_Q - R_P}{1-\cos\theta} N_P$ in step 1c. is quite intuitive. Let us consider the case where $R_P > R_Q$.When $\theta \leq \pi/2$, we displace S' to a point that is offset d from the current S' in the direction of N_Q, where d ensures that the orthogonal distances from the displaced point to tangent planes of $P(s,t)$ and $Q(u,v)$ at P^* and Q^*, respectively, are equal. That is, d is positive and must satisfy

$$R_Q + d = R_P + d\cos\theta,$$

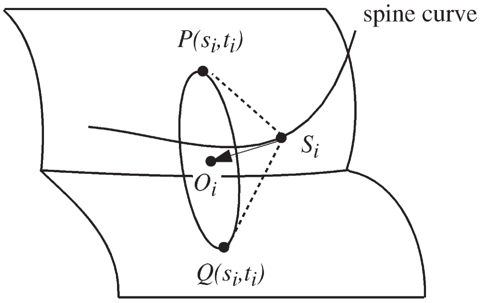
which implies that $d = \frac{R_P - R_Q}{1-\cos\theta}$ (Fig. 7a). When $\frac{\pi}{2} < \theta < \pi$, d must satisfy

$$R_Q + d = R_P - d\cos(\pi - \theta),$$

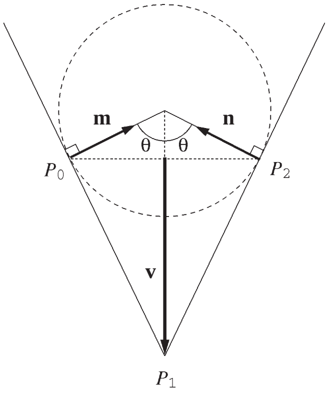
which implies also that $d = \frac{R_P - R_Q}{1-\cos\theta}$ (Fig. 7b).

6 Representing the variable-radius blends

Having derived points on the spine and linkage curves that correspond to each other, we next approximate linkage curves and the curve for centers of contributing circles with parametric curves and then sweep a circular arc along the linkage curves to construct the blend in parametric form. The variable-radius circular blend is constructed by selecting the cross-section circle as the greatest circle of the sphere that passes through the spine point and its foot points on P and Q. For the variable-radius spherical blend, the contributing circle on each sphere must lie on the plane that is perpendicular to the tangent of the spine at the center of the sphere and contains the foot points.



8



9

Fig. 8. Center of the contributing circle for spherical blends

Fig. 9. Computing the control point P_1

6.1 Centers of contributing circles

Let $(S_i, s_i, t_i, u_i, v_i, T_i)$ denote the ith output of the marching process, where S_i is the spine point, (s_i, t_i) and (u_i, v_i) are preimages of S_i on P and Q, respectively, and T_i is the spine tangent at S_i. For circular blends, we consider the contributing circle for sphere centering at S_i as the greatest circle passing through S_i, $P(s_i,t_i)$, and $Q(u_i,v_i)$. Hence the center of contributing circle, denoted by O_i, is S_i, for each i. For spherical blends, the contributing circle lies on the plane perpendicular to the spine tangent at S_i. Consequently, for each i (Fig. 8)

$$O_i = S_i + \left[(P(s_i,\, t_i) - S_i) \cdot \frac{T_i}{\|T_i\|} \right] \frac{T_i}{\|T_i\|}.$$

We call the curve formed by O_i the *center curve* of the contributing circle.

6.2 Interpolating center and linkage curves

To ensure positional continuity between the blend and base surfaces, linkage curves are first fitted in parameter spaces, namely (s,t)-space and (u,v)-space, and then represented as the mapping of the domain curves on the surfaces. Moreover, to maintain the correct point correspondence after fit-

ting, the spine and linkage curves are fitted simultaneously by regarding $O_j=(x_j,y_j,z_j)$, (s_j,t_j), and (u_j,v_j) as a single data point $(x_j,y_j,z_j,s_j,t_j,u_j,v_j)$. General methods for B-spline curve fitting are applied. The knot vector is set to be open uniform and the chord length between data points is used to approximate the parameter value β_j for $(x_j,y_j,z_j,s_j,t_j,u_j,v_j)$. The number of control points is predetermined by a bisection procedure that searches for an appropriate number of control points in a binary-search fashion. The resulting number of control points guarantees that the fitting error is within a prescribed tolerance. The number of control points is generally less than the number of data points, and hence the fitting is solved by a least-squares computation. The parameter value β_j for each data point is adjusted by Newton-Raphson iteration as described by Plass and Stone (1983).
The fitting produces the center curve

$$O(\beta)=(x(\beta),y(\beta),z(\beta))$$

and the linkage curves in the parameter spaces

$$l_P(\beta)=(s(\beta),t(\beta)) \quad \text{and} \quad l_Q(\beta)=(u(\beta),v(\beta)).$$

The linkage curves on the base surfaces are then $P(l_P(\beta))$ and $Q(l_Q(\beta))$. Note that, for variable-radius circular blends, data points of the center curve are not fitted since only surface normals at $P(l_P(\beta))$ and $Q(l_Q(\beta))$ are needed.

6.3 Constructing the variable-radius blend

The variable-radius blend is constructed by sweeping a cross-section curve along the fitted linkage curves. The cross-section curve is a circular arc represented by the following rational quadratic Bézier curve:

$$B(\alpha)=\frac{\omega_0(1-\alpha)^2P_0+2\omega_1(1-\alpha)\alpha P_1+\omega_2\alpha^2P_2}{\omega_0(1-\alpha)^2+2\omega_1(1-\alpha)\alpha+\omega_2\alpha^2}, \tag{10}$$

where P_0,P_1, and P_2 are the control points and ω_0,ω_1, and ω_2 are the weights. The control points P_0 and P_2 are located on the respective linkage curves and are expressed as

$$P_0(\beta)=P(s(\beta),t(\beta)) \;\; and \;\; P_2(\beta)=Q(u(\beta),v(\beta)).$$

As in (Choi and Ju 1989; Chuang et al. 1995), P_1 can be written as

$$P_1=\frac{1}{2}(P_0+P_2)-|P_0-P_2|\frac{\sqrt{2(1-\mathbf{m}\cdot\mathbf{n})}}{4(1+\mathbf{m}\cdot\mathbf{n})}(\mathbf{m}+\mathbf{n}).$$

For circular blends, **m** and **n** are unit normals of P and Q at $P_0(\beta)$ and $P_1(\beta)$, respectively, while, for spherical blends, we have **m** and **n** equal to the unit vectors of $O(\beta)-P_0(\beta)$ and $O(\beta)-P_2(\beta)$, respectively. The derivation is illustrated in Fig. 9.
Since the rational quadratic Bézier curve is required to be a circular arc, the weights must be suitably controlled. Let ω_0 and ω_2 be equal to 1. ω_1 must be $\cos\alpha$ and can be written in terms of **m** and **n** as

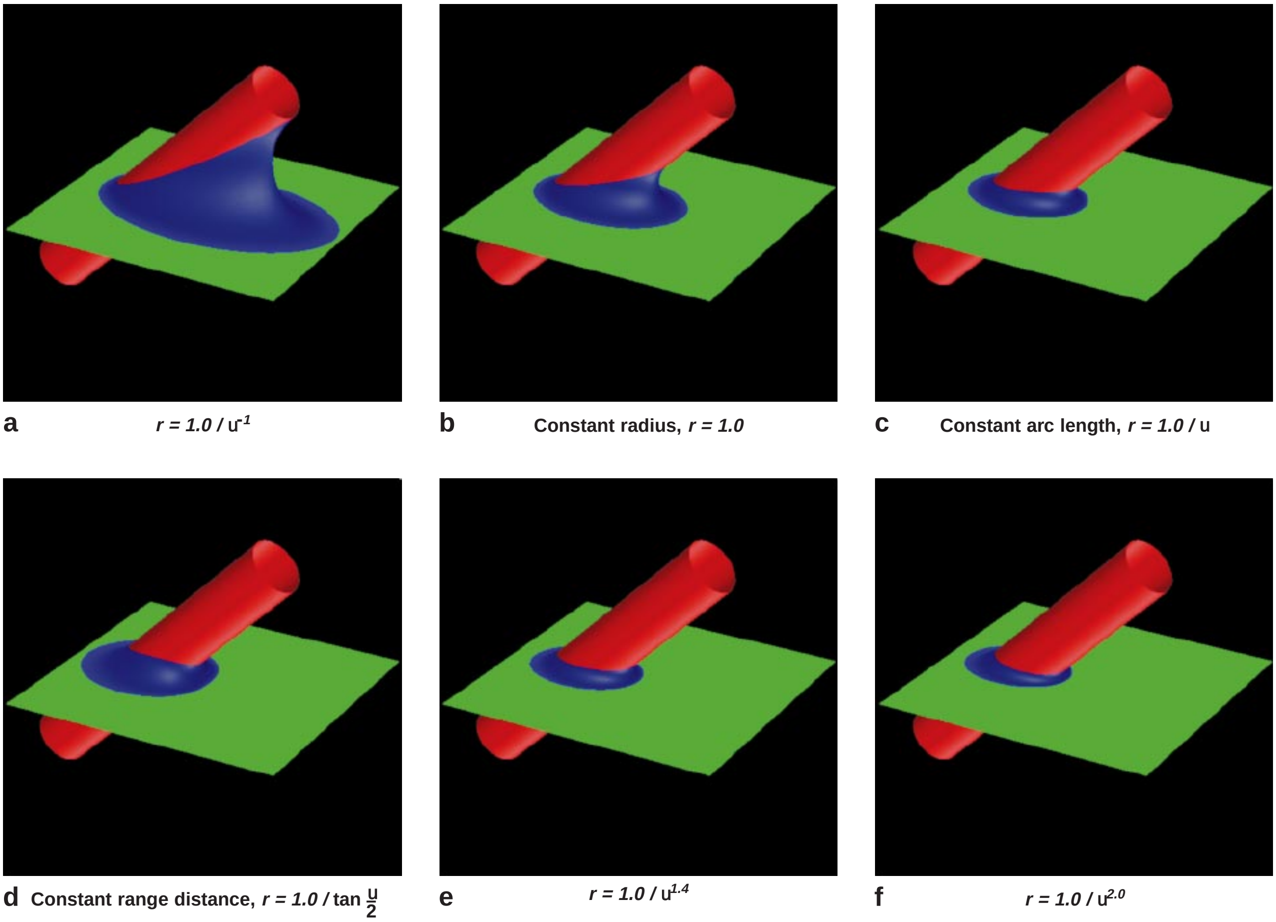
$$\omega_1=\cos\alpha=\cos\theta=\sqrt{\frac{1+\mathbf{m}\cdot\mathbf{n}}{2}}$$

where **m** and **n** are unit vectors pointing from $P_0(\beta)$ and $P_1(\beta)$, respectively, toward the circle's center. Substituting $P_0(\beta)$,$P_1(\beta)$, and $P_2(\beta)$ into Eq. 10 yields the blending surface

$$B(\alpha,\beta)=\frac{(1-\alpha)^2P_0(\beta)+2\omega_1(1-\alpha)\alpha P_1(\beta)+\alpha^2P_2(\beta)}{(1-\alpha)^2+2\omega_1(1-\alpha)\alpha+\alpha^2}. \tag{11}$$

7 Implementations and examples

The proposed constraints for the variable-radius spine and marching procedures have been implemented on an SGI Indigo2 workstation. Shown in Fig. 10 are spherical blends between a cylinder and an inclined plane using constraints $r(\theta)=1.0/\theta^{-1}$, $r(\theta)=1.0$ (constant-radius), $r(\theta)=1.0/\theta$ (constant arc length), $r(\theta)=1.0/\tan\frac{\theta}{2}$ (constant range distance), $r(\theta)=1.0/\theta^{1.4}$, and $r(\theta)=1.0/\theta^2$, respectively. Fig. 11 shows spherical blends between two intersecting Bézier surfaces using radius constraints $r(\theta)=3.0/\theta^{-1}$, $r(\theta)=3.0$, $r(\theta)=3.0/\theta^{1.4}$, and $r(\theta)=3.0/\theta^2$. These figures reveal that the proposed scheme



a $r = 1.0 / u^{-1}$

b Constant radius, $r = 1.0$

c Constant arc length, $r = 1.0 / u$

d Constant range distance, $r = 1.0 / \tan \frac{u}{2}$

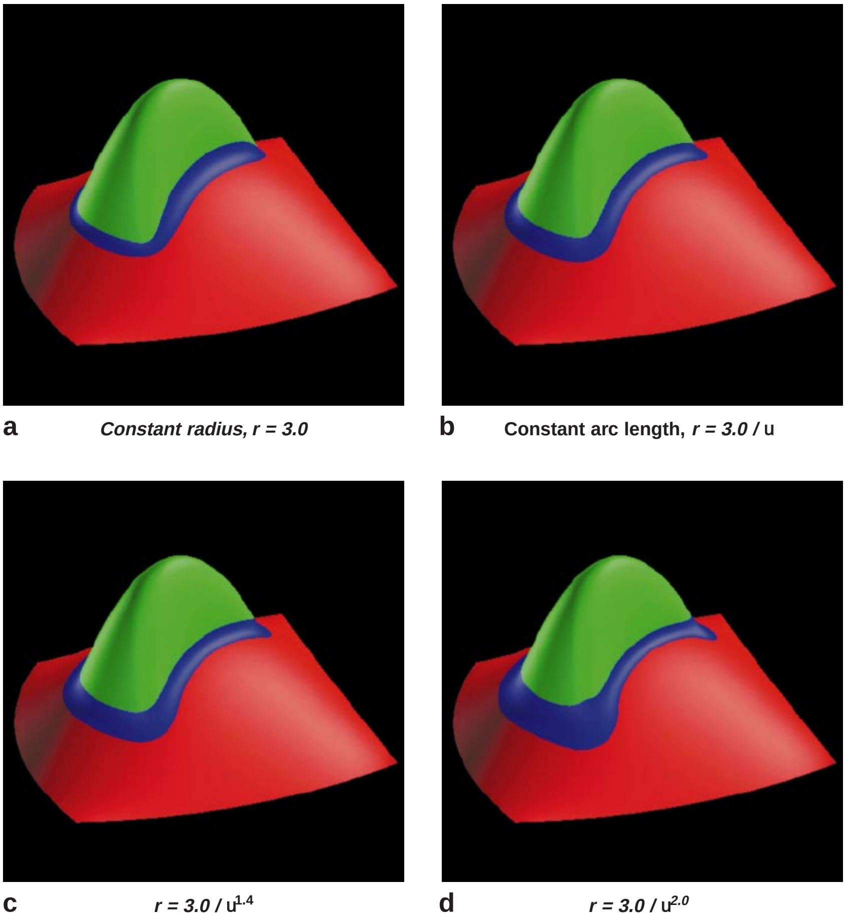
e $r = 1.0 / u^{1.4}$

f $r = 1.0 / u^{2.0}$

Fig. 10a–f. Spherical blends of a cylinder and an inclined plane

for radius constraints is able to effectively produce variable-radius blends to meet practical considerations. Our experience is that the constraint function $r(\theta) = c/\theta^p$ is simple enough and performs well. With only two parameters c and p, users are able to effectively control the shape of the blend. For examples, constant-radius constraint ($p = 0$) is appropriate for representing blends resulting from NC cutting, while constraints with $p > 1$ effectively reduce the radius in the area where two base surfaces meet in small angles.

Tables 1 and 2 compare the timing data for two tracing schemes. We observe that tracing in R^3 is much more efficient than numerical tracing in high-dimensional space using a dimensionality paradigm, since solving linear systems for the local approximant and performing point refinement with Newton iteration are computationally expensive. Moreover, a partial of Eq. 9 can be very complicated for certain constraints, which, in turn, complicates the polynomial evaluation. One observation worth mentioning is that the point refinement using Newton iteration in the high-dimensional tracing usually takes only three to four iterations on average, while the point refinement using geometric properties in the proposed 3D tracing may take on average 0.5 to 2.5 steps and 4 to 47 steps to get an equidistance point and a spine point, respectively. When using the dimensionality paradigm, tracing the spines in R^9 for variable-radius spines demands much more computing time than tracing in R^7 for constant-radius spines. In the proposed 3D tracing, all radius spines, however, require about the same order of computing time. A final special note is that base surfaces in NURBS form generally demand a higher surface manipulation cost in high-dimensional marching than polynomial base surfaces, as depicted in Tables 1 and 2



a *Constant radius, r = 3.0*

b Constant arc length, *r = 3.0 /* u

c *r = 3.0 /* u$^{1.4}$

d *r = 3.0 /* u$^{2.0}$

Fig. 11a–d. Spherical blends of two Bézier surfaces

Table 1. Times in seconds for marching the spine of a cylinder and an inclined plane

	$r(\theta)=\frac{1.0}{\theta^{-1}}$	$r(\theta)=1.0$	$r(\theta)=\frac{1.0}{\theta}$	$r(\theta)=\frac{1.0}{\tan(\theta/2)}$	$r(\theta)=\frac{1.0}{\theta^{1.4}}$	$r(\theta)=\frac{1.0}{\theta^{2}}$
3D marching	21.878	8.459	8.386	10.178	9.322	10.136
H-D marching	582.008	56.172	376.137	523.746	418.473	410.958

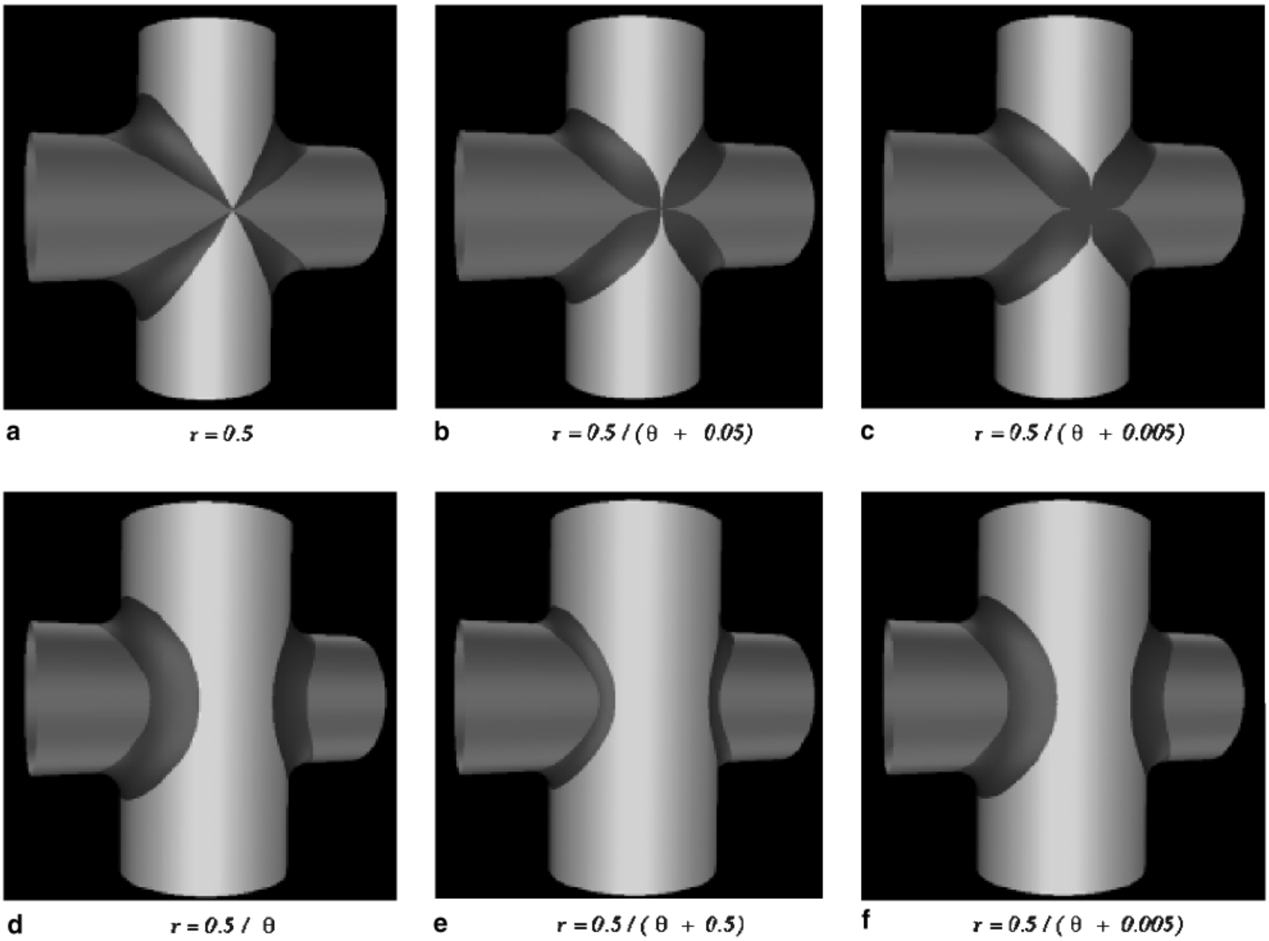
Table 2. Times in seconds for marching the spine of two Bézier surfaces

	$r(\theta)=\frac{3.0}{\theta^{-1}}$	$r(\theta)=3.0$	$r(\theta)=\frac{3.0}{\theta}$	$r(\theta)=\frac{3.0}{\tan(\theta/2)}$	$r(\theta)=\frac{3.0}{\theta^{1.4}}$	$r(\theta)=\frac{3.0}{\theta^{2}}$
3D marching	5.126	3.981	11.395	17.171	13.452	28.436
H-D marching	202.346	30.828	261.380	352.137	324.601	375.363

where cylinders are represented in NURBS form and Bézier surfaces are in polynomial form.

In practical implementations, we must deal with the cases where $f(\theta)$ approaches 0, which implies the derived radius goes to infinity and the resulting system becomes nearly singular or singular. We handle such cases by rewriting Eq. 1 as

$$r(\theta)=\frac{c}{f(\theta)+\varepsilon}, \tag{12}$$



a $r = 0.5$

b $r = 0.5 / (\theta + 0.05)$

c $r = 0.5 / (\theta + 0.005)$

d $r = 0.5 / \theta$

e $r = 0.5 / (\theta + 0.5)$

f $r = 0.5 / (\theta + 0.005)$

Fig. 12a–f. Effects of epsilon when blending two intersecting cylinders with equal and different radii

whenever $f(\theta)$ is not a constant, where ϵ is a small and fixed number. Since this modified radius constraint will be applied everywhere in the spine generation, ϵ is expected to be as small as possible to reduce its global effects. However, ϵ must be large enough to avoid the numerical instability caused by the vanishing of $f(\theta)$. We have found that the ϵ determined by the maximum allowable radius r_m using

$$r_m = \frac{c}{\varepsilon}$$

is very effective in practice. The maximum allowable radius r_m can be two or three times as large as the diameter of the base surfaces. Figure 12 depicts the effects of applying the modified radius to the blend generation. Figure 12a shows the constant-radius blend of two cylinders of identical radius. The marching of the spine with constant arc length constraint will fail at the points where normals of two cylinders coincide. Figure 12b and c shows such a blend using the modified constraint with ϵ equal to 0.05 and 0.005, respectively. The effect of modified constraints near the singular point resembles the shape of constant-radius blend, as shown in Fig. 12a. Figure 12d–f reveals the effects of applying the modified constant arc length constraint to two intersecting cylinders with different radii, in which $f(\theta)$ is nowhere to be 0. The modified constraint with a smaller ϵ makes a smaller shape deviation in the resulting blend than the original constraint.

8 Conclusion

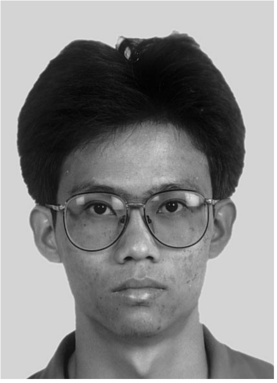
We have proposed geometric constraints that constrain the variable-radius spine and produce smooth, visually appealing variable-radius blends. The constraints can easily be adopted into a high-dimensional representation resulting in an exact form of the spine, which can be traced in high-dimensional space. Alternatively, the constraints can be implemented in a marching procedure that marches the spine and linkage curves simultaneously in R^3. Our experiments show that the proposed scheme for radius constraints is able to effectively produce variable-radius blends to meet practical considerations, and marching in R^3 is much more efficient than tracing in high-dimensional space. For future work, constraints that are capable of representing a wide range of surface blendings are of great interest.

Acknowledgements. This work was supported by the National Science Council (NSC) of the Republic of China under grant NSC 83-0408-E-009-023.

References

1. Barnhill RE, Kersey SN (1990) A marching method for parametric surface/surface intersection. Comput Aided Geom Des 7:257–280
2. Barnhill RE, Farin GE, Chen Q (1993) Constant-radius blending of parametric surfaces. In: Farin GE, Hagen H, and Noltemeier H (eds) Geometrie modeling. Springer, Berlin Heidelberg New York, pp 1–20
3. Chandru V, Dutta D, Hoffmann CM (1990) Variable radius blending using dupin cyclides. In: Wozny MJ, Turner JU, Preiss K (eds) Geometric modeling for product engineering. North-Holland, Amsterdam, pp 39–57
4. Choi BK (1991) Surface modeling for CAD/CAM. Elsevier, Amsterdam
5. Choi BK, Ju SY (1989) Constant-radius blending in surface modelling. Comput Aided Des 21:213–220
6. Chuang JH, Hoffmann CM (1992) Curvature computations on surfaces in *n*-space. Math modelling numerical anal 26:95–112
7. Chuang JH, Lien FL (1995) One and two-parameter blending for blending parametric surfaces. In: Shin SY, Kunii TL (eds) Computer graphics and applications. World Scientific, Singapore, pp 333–347
8. Chuang JH, Lin CH, Hwang WC (1995) Variable-radius blending of parametric surfaces. Visual Comput 11:513–525
9. Filip DJ (1989) Blending parametric surfaces. ACM Trans Graph 8:164–173
10. Harada T, Konno K, Chiyokura H (1991) Variable-radius blending by using Gregory patches in geometric modeling. Proceedings of Eurographics '91, North-Holland, Amsterdam, pp 507–518
11. Hoffmann CM (1990) A dimensionality paradigm for surface interrogations. Comput Aid Geom Des 7:517–532
12. Hoffmann CM, Hopcroft J (1987) The potential method for blending surfaces and coners. In: Farin G (ed) Geometric modeling: algorithms and new trends. SIAM Publications, Philadelphia, pp 347–365
13. Klass R, Kuhn B (1992) Fillet and surface intersections defined by rolling balls. Comput Aided Geom Des 9:185–193
14. Koparkar P (1991) Designing parametric blends: surface model and geometric correspondence. Visual Comput 7:39–58
15. Lee T, Bedi S, Dubey RN (1993) A parametric surface blending method for complex engineering objects. In: Rossignac J, Turner J, Allen G (eds) Proceeding of 2nd Symposium on Solid Modeling and Applications, ACM Press, New York, pp 179–188
16. Middleditch AE, Sears KH (1985) Blend surfaces for set theoretic volume modelling systems. Comput Graph 19:161–170
17. Pegna J, Wilde DJ (1990) Spherical and circual blending of functional surfaces. Trans ASME, J Offshore Mechanics Arctic Eng 112:134–142
18. Plass M, Stone M (1983) Curve-fitting with piecewise parametric curves. Comput Graph 17:229–239
19. Rockwood AP, Owen JC (1987) Blending surfaces in solid modeling. In: Farin GE (ed) Geometric modeling: algorithms and new trends. SIAM Publications, Philadelphia, pp 367–383
20. Vaishnav H, Rockwood A (1993) Blending parametric objects by implicit techniques. In: Rossignac J, Turner J, Allen G (eds) Proceeding of 2nd Symposium on Solid Modeling and Applications, ACM Press, New York, pp 165–168
21. Varady T (1989) Rolling ball blends in solid modelling. In: Kimura F, Rolstadas A (eds) Computer applications in production and engineering. Elsevier (North-Holland), Amsterdam, pp 295–308

Jung-Hong Chuang is an Associate Professor of Computer Science and Information Engineering at National Chiao Tung University, Taiwan, ROC. His research interests include geometric and solid modeling, computer graphics, and visualization. Dr. Chuang received his BS degree in Applied Mathematics from National Chiao Tung University, Taiwan, in 1978, and his MS and PhD degrees in Computer Science from Purdue University in 1987 and 1990, respectively.



Wei-Chung Hwang is a PhD student at the Department of Computer Science and Information Engineering at National Chiao Tung University, Taiwan, ROC. His research interests include geometric modeling and computer graphics. Mr. Hwang received his BS and MS degrees in Computer Science and Information Engineering from National Chiao Tung University in 1990 and 1992, respectively.