

A Method for Deforming Polygonal Shapes into Smooth Spline Surface Models

Chiew-Lan Tai, Kia-Fock Loe¹, Brian A. Barsky², and Yim-Hung Chan

Department of Computer Science

The Hong Kong University of Science and Technology

Clear Water Bay, Kowloon, Hong Kong

Abstract

This paper describes a new spline formulation that supports deformation of polygonal shapes into smooth spline surface models. Once a polygonal shape with underlying rectangular topology is specified by the user, it is deformed into a smooth surface that interpolates all the polygonal vertices. The user can then modify the default smooth surface by increasing or decreasing the amount of deformation, either globally or locally. This is accomplished by interactively controlling the shape parameters associated with the polygonal vertices. This modeling paradigm is conceptually simple, and allows C^2 continuous surfaces to be easily designed, even by a novice user.

1. Introduction

Deformation from polygonal shapes to smooth surfaces is an attractive design tool since it relieves the user from the burden of having to first specify a large set of control points and then stitching complex patches together. This paper presents a new interpolating spline formulation that supports such a modeling paradigm. It only requires the user to manipulate quadrilaterals to design polygonal shapes which are then smoothed or deformed by controlling the shape parameters associated with the vertices. The resulting surface is C^2 continuous and interpolates the given vertices. The formulation allows sharp corners to be easily introduced without using multiple knots or multiple control points.

Our deformation formulation applies only to polygonal shapes of rectangular topology. Cylindrical or toroidal topology is also supported since they can be tiled by rectangular patches. The smoothing of polygonal shapes is achieved by combining a network of *singularly reparametrized bilinear patches*[12] with a piecewise

smooth surface to obtain a new surface that interpolates all the polygonal vertices. In addition to not needing to solve a system of equations to achieve interpolation, this approach has the advantages of having local tension shape parameters and being C^2 continuous.

2. Related work

Barr [2] introduced deformation operations for twisting, stretching, bending and tapering surfaces around a central axis. His work was followed by Sederberg and Parry's [13] more generalized deformation technique, called the Free-Form Deformation (FFD) method. The FFD method embeds the object to be deformed in a lattice of control vertices that defines a trivariate Bézier volume. Deformation is done by altering the lattice and recalculating the new positions of the points of the embedded object based on their unique parameter values. Several extended versions of FFDs have since been proposed, incorporating B-spline and NURBS trivariate volumes [11] and allowing lattices of arbitrary shape and topology [9]. Other more recent deformation techniques can be found in the comprehensive survey paper of Bechmann [4]. None of them, however, addresses deformation of polygonal shapes to smooth spline surfaces.

In addition to purely geometric deformation approaches, there are physics-based deformable models. A physics-based model employs elasticity theory to construct an energy functional, and calculates the model's shape directly by finding a minimum to the energy functional or by solving a set of differential equations [6, 16]. Physics-based models can most closely mimic the sculpting effect and their optimization algorithms automatically produce fair surfaces. However, they typically require intensive computations.

To produce curved surfaces from polygonal shapes, a traditional method is to round sharp corners and edges using parametric blending and filleting techniques [7, 8]. Parametric blends typically join the base surfaces with only C^1 continuity. Another method is to replace each face of a polyhedron by an algebraic surface patch. Bajaj and Ihm[1] replace each face of a triangulated polyhedron by a low de-

¹Department of Computer Science, School of Computing, National University of Singapore, Lower Kent Ridge, Singapore 119260.

²On sabbatical leave from Computer Science Division, University of California, Berkeley, CA 94720-1776, U.S.A.

gree (5 or 7) algebraic surface patch, which has independent degrees of freedom that provide local shape control. More recently, subdivision techniques [5, 10, 14] have emerged as a popular method for obtaining smooth surfaces from polygonal meshes; this is largely due to their elegance in supporting meshes of arbitrary topology and their relationship with multiresolution techniques like wavelets.

3. Singularly reparametrized line segment

A *singularly reparametrized (SR) line segment* is a line segment that possesses parametric derivative(s) equal to zero at each end. It is obtained by blending two endpoints with a *singular blending function*[12, 15]. A function $s(t)$, $t \in [0, 1]$, is called an m th-level singular blending function if

1. it is monotonically increasing from $s(0)=0$ to $s(1)=1$
2. its first to m th derivatives are equal to zero at $t = 0$ and $t = 1$; i.e., $s^{(k)}(0) = s^{(k)}(1) = 0$ for $k = 1, \dots, m$.

One possibility for an m th-level singular blending function is $s_1(t) = 1 - (1-t^{m+1})^{m+1}$. A lower degree alternative for the same level can be developed as a special case, $s_2(t)$, of the Hermite polynomial [3]. In addition, for our purpose of formulating C^2 continuous splines, we note that the second-level singular blending function is sufficient. In this case of $m = 2$, $s_1(t) = 1 - (1-t^3)^3$ and the Hermite polynomial is $s_2(t) = 10t^3 - 15t^4 + 6t^5$. Comparing $s_1(t)$ with $s_2(t)$, the latter has a lower degree but requires slightly more computation. For the remainder of this paper, we do not assume any specific $s(t)$ function, it can be any second- or higher level singular blending function.

Using a singular blending function $s(t)$, we can blend two adjacent points \mathbf{V}_j and \mathbf{V}_{j+1} to produce an SR line segment $\mathbf{L}_j(t)$; that is,

$$\mathbf{L}_j(t) = (1 - s(t))\mathbf{V}_j + s(t)\mathbf{V}_{j+1} \quad t \in [0, 1] \quad (1)$$

which interpolates the two points as well as satisfies the following property

$$\mathbf{L}'_j(0) = \mathbf{L}'_j(1) = \mathbf{L}''_j(0) = \mathbf{L}''_j(1) = 0.$$

4. Singular blending

Given a C^2 continuous piecewise curve (comprising a sequence of curve segments), a sequence of SR line segments can be blended with it to create a new piecewise curve that preserves the continuity of the given curve. The blending introduces a parameter which can simulate the effect of tension. Denoting the given piecewise curve by $\mathbf{C}(t)$ and the sequence of SR line segments by $\mathbf{L}_j(t)$, $j = 1, \dots, n$, then the new piecewise curve $\mathbf{Q}(t)$ is

$$\mathbf{Q}_j(t) = (1 - \alpha)\mathbf{C}_j(t) + \alpha\mathbf{L}_j(t), \quad j = 1, \dots, n \quad (2)$$

where $0 \leq \alpha \leq 1$. If the SR line segments are very different from the given piecewise curve $\mathbf{C}_j(t)$, then the parameter α simply indicates the blending proportions; however, if they are a linearized approximation of the given $\mathbf{C}(t)$, then α can simulate the tension effect. Since both $\mathbf{C}(t)$ and $\mathbf{L}_j(t)$, $j = 1, \dots, n$, are C^2 continuous, thus the curve $\mathbf{Q}(t)$ must also be C^2 .

Fig. 1(a) illustrates the concept of singular blending by blending a polygon $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4$ with a smoothing unit circle $\mathbf{C}(t)$. In this case, the edges of the polygon are defined as SR line segments as specified in Eqn.(1) for $\mathbf{L}_j(t)$, $j = 1, \dots, 4$, where $\mathbf{V}_i = \mathbf{P}_i$, $i = 1, \dots, 4$, and the smoothing circle is defined by

$$\mathbf{C}_j(t) = (\cos \frac{2\pi(j-1+t)}{4}, \sin \frac{2\pi(j-1+t)}{4})$$

$t \in [0, 1]$, $j = 1, \dots, 4$. By blending the polygon with the circle using Eqn.(2), we obtain a C^2 continuous closed curve $\mathbf{Q}(t)$ which can be interpreted as a deformed or smoothed version of the polygon.

When the smoothing curve $\mathbf{C}(t)$ does not interpolate the vertices of the polygon, as is the case in this example, the resulting curve $\mathbf{Q}(t)$ may bear little resemblance to the polygonal shape for low values of α , which could be undesirable. However, there is no reason to restrict the SR line segments to be the polygon edges. Instead, we will use the following more general approach. We will determine a new sequence of SR line segments $\mathbf{L}_j(t)$, $j = 1, \dots, n$ such that when they are blended with the smoothing curve $\mathbf{C}(t)$, the resulting curve $\mathbf{Q}(t)$ will interpolate the polygon vertices.

This is shown in Fig. 1(b). Given the smoothing circle $\mathbf{C}(t)$, and the polygon $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4$, we can find a new sequence of SR line segments $\mathbf{L}_j(t)$, $j = 1, \dots, n$ which when blended with the smoothing circle produces the curve $\mathbf{Q}(t)$ that interpolates the polygon vertices.

This approach can be extended to smoothing polygonal surfaces. The smoothed surface will interpolate the polygon vertices and is thus guaranteed to resemble the polygonal shapes.

This is the central idea of our approach. The user first sketches the polygonal outline of the object to be modeled in terms of nonplanar quadrilaterals without regard to smoothness of the surface. The resulting smooth surface is guaranteed to interpolate all the polygon vertices. This provides an intuitive modeling approach for the user, and should be especially appropriate for novice designers. Our modeling system initializes the α value to a reasonable default value, such as 0.5. The user can then modify the default smooth surface by varying the α value. To provide further and more precise control over the shape, we also provide local tension parameters. This is described in the next section.

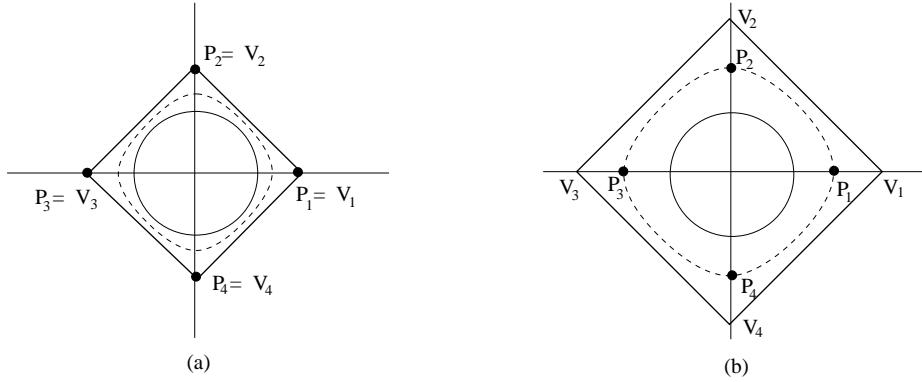


Figure 1. Blending a polygon with a smoothing circle.

5. Smoothing closed polygon

In the previous section, we gave an example of using a unit circle to smooth a closed polygon. For the technique to be a useful modeling tool, it must be improved in two ways.

First, to enable the user to adjust the tension locally at each joint, we need to generalize the constant α value in Eqn.(2) to a function of t . To provide local deformation control, we assign a local tension parameter α_j^* to each vertex \mathbf{P}_j , $j = 1, \dots, n$ and then interpolate these tension parameters using the singular blending function:

$$\alpha_j(t) = (1 - s(t))\alpha_j^* + s(t)\alpha_{j+1}^* \quad j = 1, \dots, n. \quad (3)$$

By replacing the constant α value in Eqn.(2) by the tension function $\alpha_j(t)$, we obtain a locally deformable curve $\mathbf{Q}(t)$ as follows:

$$\mathbf{Q}_j(t) = (1 - \alpha_j(t))\mathbf{C}_j(t) + \alpha_j(t)\mathbf{L}_j(t) \quad j = 1, \dots, n. \quad (4)$$

Second, we need to generalize the unit circle to a smoothing closed curve that is independent of the coordinate system and resembles the polygonal shape. We will use a B-spline to define the smoothing curve since the B-spline has many desirable geometric and computational properties.

Let $\{\mathbf{P}_1, \dots, \mathbf{P}_n\}$ be the vertices of a given n -sided polygon and $\{\alpha_1^*, \dots, \alpha_n^*\}$ be the set of corresponding tension parameters. Our approach first obtains a closed uniform cubic B-spline curve $\mathbf{C}_j(t)$ which approximates the given vertices. To produce the closed curve, three additional vertices \mathbf{P}_0 , \mathbf{P}_{n+1} , and \mathbf{P}_{n+2} are introduced, where

$$\mathbf{P}_0 = \mathbf{P}_n \quad \mathbf{P}_{n+1} = \mathbf{P}_1 \quad \mathbf{P}_{n+2} = \mathbf{P}_2.$$

Using \mathbf{P}_j , $j = 0, \dots, n + 2$ as the control points of the B-spline curve, we obtain

$$\mathbf{C}_j(t) = N_0(t)\mathbf{P}_{j-1} + N_1(t)\mathbf{P}_j + N_2(t)\mathbf{P}_{j+1} + N_3(t)\mathbf{P}_{j+2} \quad (5)$$

where $j = 1, \dots, n$ and $N_i(t)$, $i = 0, \dots, 3$ are the cubic uniform B-spline basis functions [3].

Let $\{\mathbf{V}_1, \dots, \mathbf{V}_n\}$ denote the unknown points that define the sequence of SR line segments $\mathbf{L}_j(t)$, $j = 1, \dots, n$ to be blended with the B-spline curve obtained. The SR line segments that are determined must be such that the blended curve $\mathbf{Q}(t)$ in Eqn.(4) interpolates the polygon vertices. That is, we impose the following n constraints:

$$\mathbf{P}_j = \mathbf{Q}_j(0) \quad j = 1, \dots, n.$$

Substituting for $\mathbf{Q}_j(0)$ from Eqn (4),

$$\mathbf{P}_j = (1 - \alpha_j(0))\mathbf{C}_j(0) + \alpha_j(0)\mathbf{L}_j(0) \quad j = 1, \dots, n.$$

Substituting for $\alpha_j(0)$ from Eqn.(3) and for $\mathbf{L}_j(0)$ from Eqn.(1) yields

$$\mathbf{P}_j = (1 - \alpha_j^*)\mathbf{C}_j(0) + \alpha_j^*\mathbf{V}_j \quad j = 1, \dots, n.$$

Due to the linearity of the SR line segments, the points defining them are independent of each other and hence can be determined by simple computation. Solving Eqn.(5) for \mathbf{V}_j yields:

$$\mathbf{V}_j = \mathbf{P}_j + \frac{1 - \alpha_j^*}{\alpha_j^*}(\mathbf{P}_j - \mathbf{C}_j(0)) \quad j = 1, \dots, n.$$

Note that the geometric interpretation of this equation is that when $\alpha_j^* = 1$, then $\mathbf{V}_j = \mathbf{P}_j$, and as α_j^* decreases to 0, \mathbf{V}_j moves away from \mathbf{P}_j . When $\alpha_j^* = 0$, \mathbf{V}_j is at infinity, implying no suitable SR line segments can be found. We therefore restrict the value of α_j^* to be in the range $(0, 1]$.

Since $\mathbf{L}_j(t)$, $\mathbf{C}_j(t)$, and $\alpha_j(t)$ are all C^2 continuous, the resulting curve in Eqn.(4) is clearly C^2 continuous. Differentiating Eqn.(4), we note that the end tangents are in the same directions as those in the B-spline counterpart, with the magnitudes scaled by $1 - \alpha_j^*$; that is,

$$\mathbf{Q}'_{j-1}(1) = (1 - \alpha_j^*)\mathbf{C}'_{j-1}(1) \text{ and } \mathbf{Q}'_j(0) = (1 - \alpha_j^*)\mathbf{C}'_j(0)$$

and since $\mathbf{C}'_{j-1}(1) = \mathbf{C}'_j(0)$, we have $\mathbf{Q}'_{j-1}(1) = \mathbf{Q}'_j(0)$.

When α_j^* approaches 1, the corresponding SR line segments that are determined are close to the given polygon. In addition, since the contribution of the SR line segments to the curve $\mathbf{Q}(t)$ is larger as α_j^* approaches unity, the resulting curve segments increasingly resemble the SR line segments and hence are more tensed. In the case when $\alpha_j^* = 1$ for all j , we obtain the given polygon.

Fig. 2 shows the effects of decreasing the tension parameters globally from 1 to 0.1. It illustrates the deformation of a square to rounder closed curves. To illustrate the local effect of tension parameters, Fig. 3 shows the same square with all vertices having $\alpha_j^* = 0.5$ except for the lower right one where the local tension parameter decreases from 1 to 0.25.

6 Smoothing open polygon

In the previous section, we have assumed that the given vertices $\{\mathbf{P}_1, \dots, \mathbf{P}_n\}$ form a closed polygon. For the case where these vertices form an open polygon, the polygon must be smoothed by an open B-spline curve. Since the polygon has $n - 1$ segments, to obtain the same number of segments in the open uniform cubic B-spline curve, we introduce two phantom vertices $\mathbf{P}_0, \mathbf{P}_{n+1}$ where

$$\mathbf{P}_0 = 2\mathbf{P}_1 - \mathbf{P}_2 \quad \text{and} \quad \mathbf{P}_{n+1} = 2\mathbf{P}_n - \mathbf{P}_{n-1}.$$

The phantom vertices are chosen so that the B-spline curve $\mathbf{C}_j(t), j = 1, \dots, n - 1$ interpolates the endpoints \mathbf{P}_1 and \mathbf{P}_n [3]. The interpolation constraints imposed on $\mathbf{Q}(t)$ then become

$$\mathbf{P}_j = \mathbf{Q}_j(0), j = 1, \dots, n - 1 \quad \text{and} \quad \mathbf{P}_n = \mathbf{Q}_{n-1}(1).$$

Thus, the vertices of the SR line segments can be solved from

$$\begin{aligned} \mathbf{V}_1 &= \mathbf{P}_1 & \mathbf{V}_n &= \mathbf{P}_n \\ \mathbf{V}_j &= \mathbf{P}_j + \frac{1 - \alpha_j^*}{\alpha_j^*} (\mathbf{P}_j - \mathbf{C}_j(0)) & j &= 2, \dots, n - 1. \end{aligned}$$

7. Singularly reparametrized bilinear patch

Next, we will generalize the above idea from curve to surface design. We first introduce the *singularly reparametrized (SR) bilinear patch*, which is geometrically equivalent to a bilinear surface but has vanishing end derivatives. That is, an SR bilinear patch is defined by four vertices $\{\mathbf{V}_{i,j}, \mathbf{V}_{i,j+1}, \mathbf{V}_{i+1,j}, \mathbf{V}_{i+1,j+1}\}$:

$$\begin{aligned} \mathbf{L}_{i,j}(u, v) &= (1 - s(u))(1 - s(v))\mathbf{V}_{i,j} + (1 - s(u))s(v)\mathbf{V}_{i,j+1} \\ &\quad + s(u)(1 - s(v))\mathbf{V}_{i+1,j} + s(u)s(v)\mathbf{V}_{i+1,j+1} \end{aligned} \quad (6)$$

Differentiating $\mathbf{L}_{i,j}(u, v)$ in Eqn.(6) and evaluating at $u = 0, u = 1, v = 0$, and $v = 1$, yields the following singular properties:

$$\begin{aligned} \partial_u \mathbf{L}_{i,j}(0, v) &= \partial_u \mathbf{L}_{i,j}(1, v) = \partial_v \mathbf{L}_{i,j}(u, 0) = \partial_v \mathbf{L}_{i,j}(u, 1) = 0, \\ \partial_u^2 \mathbf{L}_{i,j}(0, v) &= \partial_u^2 \mathbf{L}_{i,j}(1, v) = \partial_v^2 \mathbf{L}_{i,j}(u, 0) = \partial_v^2 \mathbf{L}_{i,j}(u, 1) = 0, \\ \partial_{uv}^2 \mathbf{L}_{i,j}(0, v) &= \partial_{uv}^2 \mathbf{L}_{i,j}(1, v) = \partial_{uv}^2 \mathbf{L}_{i,j}(u, 0) = \partial_{uv}^2 \mathbf{L}_{i,j}(u, 1) = 0. \end{aligned}$$

8. Smoothing polygonal surfaces

We now apply the idea to the smoothing of polygonal surfaces, in particular quadrilaterals. Let $\{\mathbf{P}_{i,j} : i = 1, \dots, m; j = 1, \dots, n\}$ be a network of vertices. Let $\{\alpha_{i,j}^* : i = 1, \dots, m; j = 1, \dots, n\}$ be the corresponding set of tension parameters, where $0 < \alpha_{i,j}^* \leq 1$. The network can have a rectangular, cylindrical, or toroidal topology.

First, we assume that the network has a cylindrical topology, where i is the index in the closed direction and j is the index in the open-ended direction. That is, the network of vertices has $m \times (n - 1)$ quadrilaterals.

To obtain a B-spline surface $\mathbf{S}_{i,j}(u, v), i = 1, \dots, m$ and $j = 1, \dots, n - 1$, from the network of vertices, we first introduce some additional points as in the case of curves. Analogous to handling open curves, we add two rows of phantom vertices $\mathbf{P}_{0,j}$ and $\mathbf{P}_{m+1,j}$ where $j = 1, \dots, n$; that is,

$\mathbf{P}_{0,j} = 2\mathbf{P}_{1,j} - \mathbf{P}_{2,j}$ and $\mathbf{P}_{m+1,j} = 2\mathbf{P}_{m,j} - \mathbf{P}_{m-1,j}$ $j = 1, \dots, n$. Analogous to handling closed curves, we add three rows of vertices to wrap around the end vertices

$$\mathbf{P}_{i,0} = \mathbf{P}_{i,n} \quad \mathbf{P}_{i,n+1} = \mathbf{P}_{i,1} \quad \mathbf{P}_{i,n+2} = \mathbf{P}_{i,2} \quad i = 0, \dots, m + 1.$$

The B-spline surface is then

$$\mathbf{S}_{i,j}(u, v) = \sum_{p=0}^3 \sum_{q=0}^3 N_p(u) N_q(v) \mathbf{P}_{i+p-1,j+q-1} \quad (7)$$

$i = 1, \dots, m; j = 1, \dots, n - 1$. Let $\mathbf{L}_{i,j}(u, v), i = 1, \dots, m, j = 1, \dots, n - 1$ be the unknown SR bilinear patches to be blended with the B-spline surface. The blended smooth surface is given by

$$\mathbf{Q}_{i,j}(u, v) = (1 - \alpha_{i,j}(u, v))\mathbf{S}_{i,j}(u, v) + \alpha_{i,j}(u, v)\mathbf{L}_{i,j}(u, v) \quad (8)$$

where

$$\begin{aligned} \alpha_{i,j}(u, v) &= (1 - s(u))(1 - s(v))\alpha_{i,j}^* + (1 - s(u))s(v)\alpha_{i,j+1}^* \\ &\quad + s(u)(1 - s(v))\alpha_{i+1,j}^* + s(u)s(v)\alpha_{i+1,j+1}^* \end{aligned} \quad (9)$$

$$i = 1, \dots, m; j = 1, \dots, n - 1.$$

To find the points $\{\mathbf{V}_{i,j} : i = 1, \dots, m; j = 1, \dots, n\}$ that define the SR bilinear patches, we impose a set of constraints on $\mathbf{Q}(u, v)$ to interpolate the polygonal vertices:

$$\begin{aligned} \mathbf{P}_{i,j} &= \mathbf{Q}_{i,j}(0, 0) & i &= 1, \dots, m; j = 1, \dots, n - 1, \\ \mathbf{P}_{i,n} &= \mathbf{Q}_{i,n-1}(0, 1) & i &= 1, \dots, m. \end{aligned}$$

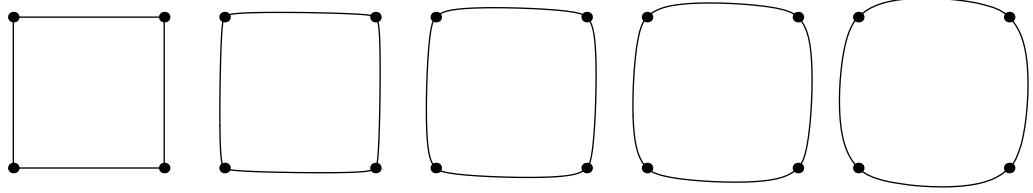


Figure 2. Decreasing the tension parameters globally from 1, 0.75, 0.5, 0.25 to 0.1.

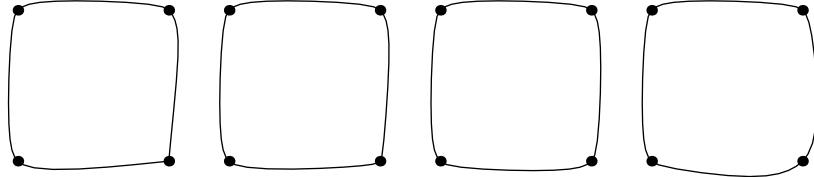


Figure 3. Decreasing the tension parameter locally (at the lower right vertex) from 1, 0.75, 0.5 to 0.25. All the other vertices have $\alpha_j^* = 0.5$.

Substituting for $\mathbf{Q}_{i,j}(0,0)$ and $\mathbf{Q}_{i,n-1}(0,1)$ from Eqn.(8), we have

$$\begin{aligned}\mathbf{P}_{i,j} &= (1 - \alpha_{i,j}(0,0))\mathbf{S}_{i,j}(0,0) + \alpha_{i,j}(0,0)\mathbf{L}_{i,j}(0,0) \\ i &= 1, \dots, m; j = 1, \dots, n-1, \\ \mathbf{P}_{i,n} &= (1 - \alpha_{i,n-1}(0,1))\mathbf{S}_{i,n-1}(0,1) \\ &\quad + \alpha_{i,n-1}(0,1)\mathbf{L}_{i,n-1}(0,1) \quad i = 1, \dots, m.\end{aligned}$$

Substituting for $\alpha_{i,j}(0,0)$ and $\alpha_{i,n-1}(0,1)$ from Eqn.(9) and for $\mathbf{L}_{i,j}(0,0)$ and $\mathbf{L}_{i,n-1}(0,1)$ from Eqn.(6), we have

$$\begin{aligned}\mathbf{P}_{i,j} &= (1 - \alpha_{i,j}^*)(\mathbf{S}_{i,j}(0,0) + \alpha_{i,j}^*\mathbf{V}_{i,j}) \\ i &= 1, \dots, m; j = 1, \dots, n-1, \\ \mathbf{P}_{i,n} &= (1 - \alpha_{i,n}^*)\mathbf{S}_{i,n-1}(0,1) + \alpha_{i,n}^*\mathbf{V}_{i,n} \\ i &= 1, \dots, m.\end{aligned}\tag{10}$$

Substituting $\mathbf{S}_{i,1}(0,0) = \mathbf{P}_{i,1}$ and $\mathbf{S}_{i,n-1}(0,1) = \mathbf{P}_{i,n}$, $i = 1, \dots, m$, and simplifying yields

$$\mathbf{V}_{i,1} = \mathbf{P}_{i,1} \quad \mathbf{V}_{i,n} = \mathbf{P}_{i,n} \quad i = 1, \dots, m.$$

The remaining $\mathbf{V}_{i,j}$ can be solved from Eqn.(10) as follows:

$$\mathbf{V}_{i,j} = \mathbf{P}_{i,j} + \frac{1 - \alpha_{i,j}^*}{\alpha_{i,j}^*}(\mathbf{P}_{i,j} - \mathbf{S}_{i,j}(0,0))$$

$i = 1, \dots, m; j = 2, \dots, n-1$. Note that the equations for computing $\mathbf{V}_{i,j}$ are very simple. The computation that is needed is much simpler than many surface interpolation approaches which require solving a system of linear equations.

We have assumed a cylindrical topology; the case of a rectangular topology can be handled in a similar way. Since both the u and v parametric directions are open-ended, we add two rows of phantom vertices in both directions. The points defining the $(m-1) \times (n-1)$ SR bilinear patches are

$$\begin{aligned}\mathbf{V}_{i,1} &= \mathbf{P}_{i,1} & \mathbf{V}_{i,n} &= \mathbf{P}_{i,n} & i &= 1, \dots, m, \\ \mathbf{V}_{1,j} &= \mathbf{P}_{1,j} & \mathbf{V}_{m,j} &= \mathbf{P}_{m,j} & j &= 1, \dots, n, \\ \mathbf{V}_{i,j} &= \mathbf{P}_{i,j} + \frac{1 - \alpha_{i,j}^*}{\alpha_{i,j}^*}(\mathbf{P}_{i,j} - \mathbf{S}_{i,j}(0,0)) \\ i &= 2, \dots, m-1; j = 2, \dots, n-1.\end{aligned}$$

9. Results

To illustrate the proposed modeling paradigm, we will now show how a vase can be modeled using our prototype system. The user first specifies a polygonal shape, as shown in Fig. 4(a). Our system then uses these vertices to define a smooth B-spline surface, which is shown in Fig. 4(b). This B-spline surface only approximates the polygonal shape. Next, the system assumes a global tension value of $\alpha = 0.5$, and determines a set of SR bilinear patches (Fig. 4(c)) to be blended with the B-spline surface to produce the smooth vase shown in Fig. 4(d). This smooth surface is C^2 continuous and interpolates all the given vertices (the black squares shown in the figure).

Fig. 5 and 6 show some possible ways of modifying the default shape ($\alpha = 0.5$) generated by the system. The left-most vase in all these figures is the default shape. Fig. 5

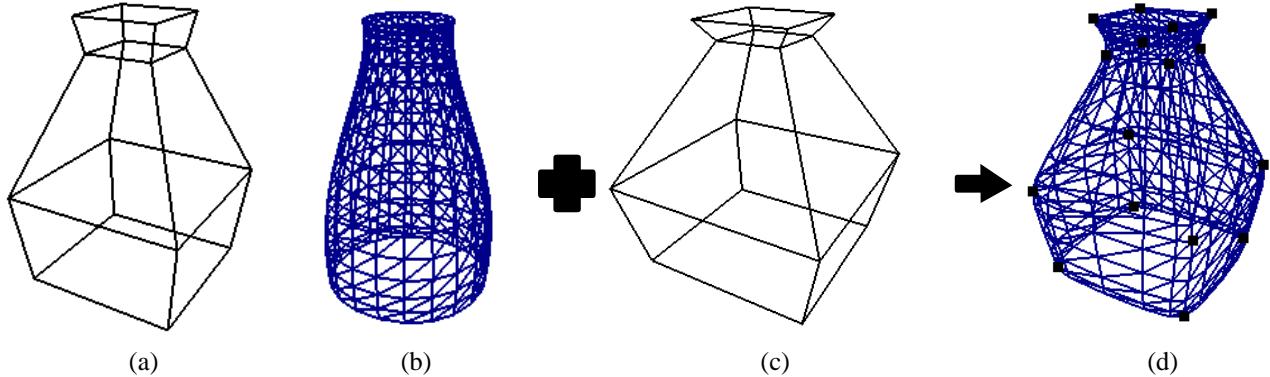


Figure 4. A user-input polygonal shape (a), then the B-spline surface (b) and the SR bilinear patches (c) are blended together using $\alpha = 0.5$ to generate vase (d). The black squares in (d) indicate the polygonal vertices in (a).

shows the effects of changing the tension parameters globally; the vase can be made to have sharper corners (Fig. 5(a)) or be rounder (Fig. 5(b)). The user can also modify the vase locally. Fig. 6 shows the effects of varying the tension parameters of those vertices in the third row.

10. Conclusion

We have presented a new spline formulation for deforming polygonal shapes to smooth surfaces. It is conceptually simple and easy to use even for novice users. The user needs only to specify a polygonal shape and then the modeling system will generate a default smooth surface interpolating all the polygonal vertices. The user can then modify the default smooth surface, either globally or locally, by changing the local tension parameters. The method is computationally simple, and guarantees C^2 continuity. The usefulness of this new modeling paradigm has been demonstrated by several modeling examples.

Acknowledgement

This work is partially supported by the Hong Kong RGC Grant DAG97/98.EG14.

References

- [1] C. L. Bajaj and I. Ihm. Smoothing polyhedra using implicit algebraic splines. *ACM SIGGRAPH '92 Proceedings*, 26(2):79–87, 1992.
- [2] A. H. Barr. Global and local deformations of solid primitives. *Computer Graphics*, 18(3):21–30, July 1984.
- [3] R. H. Bartels, J. C. Beatty, and B. A. Barsky. *An introduction to splines for use in computer graphics and geometric modeling*. Morgan Kaufmann, San Francisco, 1987.
- [4] D. Bechmann. Multidimensional free-form deformation tools. In *Eurographics '98 (State of The Art Report)*, pages 105–112, 1998.
- [5] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, Nov. 1978.
- [6] G. Celniker and D. Gossard. Deformable curve and surface finite-elements for free-form shape design. *Computer Graphics*, 25(4):257–266, 1991.
- [7] H. Chiyokura and F. Kimura. Design of solids with free-form surfaces. *ACM SIGGRAPH '83 Proceedings*, pages 289–298, 1983.
- [8] B. K. Choi and S. Y. Ju. Constant radius blending in surface modeling. *Computer-Aided Design*, 21:213–220, 1989.
- [9] S. Coquillart. Extended free-form deformation: A sculpturing tool for 3D geometric modeling. *Computer Graphics*, 24(4):187–193, Aug. 1990.
- [10] D. Doo and M. A. Sabin. Behaviour of recursive subdivision surfaces near extraordinary points. *Computer-Aided Design*, 10:356–360, 1978.
- [11] W. M. Hsu, J. F. Hughes, and H. Kaufman. Direct manipulation of free-form deformations. *ACM SIGGRAPH '92 Proceedings*, 26(2):177–184, July 1992.
- [12] K. F. Loe. α -B-spline: a linear singular blending B-spline. *The Visual Computer*, 12:18–25, 1996.
- [13] T. Sederberg and S. Parry. Free-form deformation of solid geometric models. *ACM SIGGRAPH '86 Proceedings*, 20(4):151–160, July 1986.
- [14] T. W. Sederberg, J. Zheng, D. Sewell, and M. Sabin. Non-uniform recursive subdivision surfaces. *ACM SIGGRAPH Proceedings '98*, pages 387–394, 1998.
- [15] C. L. Tai and K. F. Loe. α -spline: a C^2 continuous spline with weights and tension control. In A. Pasko, editor, *Shape Modeling International '99*, pages 138–143, Aizu-Wakamatsu, Japan, 1999. IEEE Computer Society Press.
- [16] J. A. Thingvold and E. Cohen. Physical modeling with B-spline surfaces for interactive design and animation. *Computer Graphics*, 24(4):129–137, 1990.



(a) Increasing the tension parameter globally from 0.5 (left), 0.75 (center) to 1 (right).



(b) Decreasing the tension parameter globally from 0.5 (left), 0.35 (center) to 0.25 (right).

Figure 5. Starting from the default shape ($\alpha = 0.5$), the vase can be modified to become either more tensed (a) or rounder (b).



(a) The tension parameters of the third row of vertices are increased from 0.5 (left), 0.75 (center) to 1 (right).



(b) The tension parameter of the third row of vertices are decreased from 0.5 (left), 0.35 (center) to 0.25 (right).

Figure 6. Starting from the default shape ($\alpha = 0.5$), the lower part of the vase can be modified to become either more tensed (a) or rounder (b).