Plus Curves and Surfaces

A. Ardeshir Goshtasby

Department of Computer Science and Engineering
Wright State University, Dayton, OH 45435

Abstract

The formulations for parametric curves and surfaces that are based on control points are revised to use control lines and control planes instead. Curves defined by control lines are called *control-line curves* or *plus curves*, and surfaces defined by control planes are called *control-plane surfaces* or *plus surfaces*; plus implies that in addition to the control points, tangents at the control points are used. The new curve and surface formulations provide more flexibility in 2-D and 3-D shape design than traditional formulations. Properties of plus curves and surfaces are presented and application of plus surfaces in parametric representation of irregular polygon meshes is described.

Keywords: Bézier curve; Bézier surface; rational Gaussian surface; polygon mesh; control point; control line; control plane

1 Introduction

Parametric curves and surfaces provide the basis for many geometric modeling systems. In the existing curve and surface formulations, a shape is designed by moving the control points of a curve or a surface. This paper shows that if, instead of control points, control lines and control planes are used, more effective tools for the design of free-form shapes will be obtained.

The idea of replacing control points with control lines and control planes will be demonstrated in the context of Bézier curves and surfaces, but the idea is general and can be applied to all curves and surfaces that are formulated in terms of control points. Control-line Bézier curves will be called *Bézier$^+$ curves*, and control-plane Bézier surfaces will be called *Bézier$^+$ surfaces*.

In the following sections, first the formulation for Bézier$^+$ curves is given and its properties are compared with those of Bézier curves. Then, the formulation for Bézier$^+$ surfaces is given and its properties are compared with those of Bézier surfaces. Finally, rational Gaussian (RaG) surfaces are extended to RaG$^+$ surfaces and applied to smooth parametric representation of polygon meshes.

2 Bézier and Bézier$^+$ Curves

A Bézier curve of degree k is defined by [1, 11],

$$\mathbf{P}(u) = \sum_{i=0}^{k} \mathbf{V}_i B_{i,k}(u), \tag{1}$$

where $\{\mathbf{V}_i : i = 0, ..., k\}$ is an ordered set of control points,

$$B_{i,k}(u) = \binom{k}{i} u^i (1-u)^{k-i} \tag{2}$$

is the ith Bernstein polynomial of degree k, and in 2-D, $\mathbf{V}_i = \left[\begin{smallmatrix} x_i \\ y_i \end{smallmatrix}\right]$. Since the polynomials are predefined, once the control points are selected the curve becomes fixed. Bézier curves pass through the end control points and are tangent to the end control-polygon edges. Moreover, they are affine invariant, fall inside the convex hull of the control points, and do not intersect a straight line more than their control polygons would.

To obtain a Bézier$^+$ curve, the control points in formula (1) are replaced with control lines:

$$\mathbf{P}^+(u) = \sum_{i=0}^{k} \mathbf{L}_i(u) B_{i,k}(u). \tag{3}$$

In 2-D,

$$\mathbf{L}_i(u) = \begin{bmatrix} lx_i(u) \\ ly_i(u) \end{bmatrix}, \tag{4}$$

and so,

$$\mathbf{P}^+(u) = \begin{bmatrix} x^+(u) \\ y^+(u) \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{k} lx_i(u) B_{i,k}(u) \\ \sum_{i=0}^{k} ly_i(u) B_{i,k}(u) \end{bmatrix}. \tag{5}$$

$lx_i(u)$ is a line that passes through point (u_i, x_i) and has a slope, which we denote by $x_i^{'}$, and u_i is the ith parameter coordinate of the curve. Therefore,

$$lx_i(u) = x_i^{'} u + b_i. \tag{6}$$

b_i is determined such that the line with slope $x_i^{'}$ will pass through point (u_i, x_i). That is,

$$b_i = x_i - x_i^{'} u_i. \tag{7}$$

Therefore, we have

$$x^+(u) = \sum_{i=0}^{k} x_i B_{i,k}(u) + \sum_{i=0}^{k} x_i^{'} (u - u_i) B_{i,k}(u). \tag{8}$$

Similarly, we find

$$y^+(u) = \sum_{i=0}^{k} y_i B_{i,k}(u) + \sum_{i=0}^{k} y_i^{'} (u - u_i) B_{i,k}(u). \tag{9}$$

Let's denote the tangent vector associated with the ith control point by $\mathbf{P}_i' = \begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = s \begin{bmatrix} X_i' \\ Y_i' \end{bmatrix}$, where s is the tangent magnitude and $\begin{bmatrix} X_i' \\ Y_i' \end{bmatrix}$ is the tangent direction. Although the tangent magnitude associated with each control point can be specified independently, we denote all tangent magnitudes by s to minimize the number of free parameters the designer has to work with. Therefore, a Bézier$^+$ curve in 2-D is defined by:

$$x^+(u) = \sum_{i=0}^{k} x_i B_{i,k}(u) + s \sum_{i=0}^{k} X_i'(u - u_i) B_{i,k}(u), \tag{10}$$

$$y^+(u) = \sum_{i=0}^{k} y_i B_{i,k}(u) + s \sum_{i=0}^{k} Y_i'(u - u_i) B_{i,k}(u). \tag{11}$$

Given the control points of a curve, we initially estimate the tangent directions from

$$X_i' = \frac{Vx_{i+1} - Vx_{i-1}}{\|\mathbf{V}_{i+1} - \mathbf{V}_{i-1}\|}, \tag{12}$$

$$Y_i' = \frac{Vy_{i+1} - Vy_{i-1}}{\|\mathbf{V}_{i+1} - \mathbf{V}_{i-1}\|}, \tag{13}$$

where Vx_i and Vy_i are the x and y components of $\mathbf{V}_i$. At the end points, we replace $\mathbf{V}_{-1}$ with $\mathbf{V}_0$, and $\mathbf{V}_{k+1}$ with $\mathbf{V}_k$ when finding the tangent directions. We also initially let $s = 1$ and let the user interactively revise it as needed. A Bézier curve obtained from four control points is shown in Fig. 1a. Bézier$^+$ curves obtained from the same set of control points with different tangent magnitudes are shown in Figs. 1b–d. When the tangent magnitude at the control points is 0, the obtained Bézier$^+$ curve will be the same as the Bézier curve.

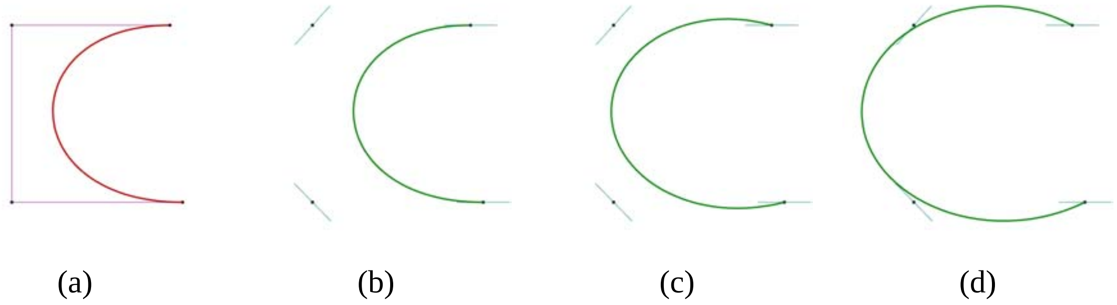


Figure 1: (a) A Bézier curve. (b)–(d) Bézier$^+$ curves obtained with increasing tangent magnitudes.

The tangent directions enable revision of a curve locally as demonstrated in Fig. 2. Figs. 2a–c are all obtained from the same set of control points, the same tangent magnitudes, and the same tangent directions at the end control points, but from different tangent directions at the two interior control points.

The formula for a Bézier$^+$ curve consists of two terms (see equations (10) and (11)). The first term defines a regular Bézier curve and shows the contribution of the control points on the

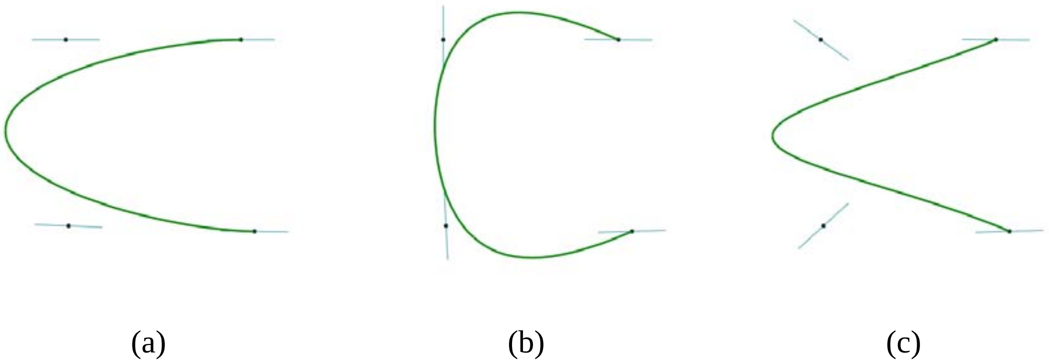


Figure 2: (a)–(c) Bézier$^+$ curves obtained from the same control points, the same tangent magnitudes, and the same tangent directions at the end control points, but from different tangent directions at the two interior control points.

obtained curve. The second term shows the contribution of the tangent vectors on the curve. The tangent magnitude can be considered the relative importance of tangent directions with respect to the control points in shape design. When this weight is zero, tangent directions will not have any effect on the obtained curve; therefore, the obtained curve becomes the same as the Bézier curve. However, as this weight is increased, the influence of the tangent vectors on the curve increases and the curve expands.

A Bézier$^+$ curve interpolates its first and last control points. This is obvious because the first term in equations (10) and (11) show a regular Bézier curve, and the second term in these equations become zero at $u = 0$ and $u = 1$. A Bézier$^+$ curve, in general, is not tangent to the first and last control lines. However, the curve can be made to become tangent to the first and last control lines by adjusting the tangent directions at two of the interior control points. For instance, for the curve to become tangent to the first control line, the first derivatives of (10) and (11) are set to $X_0^{'}$ and $Y_0^{'}$ and the obtained equations are solved for $X_1^{'}$ and $Y_1^{'}$. This will determine the tangent direction at the second control point. Figs. 3a and 3b show Bézier$^+$ curves where the two interior tangent directions are determined such that the curve becomes tangent to the first and last control lines. If in addition, the first and last control points coincide, a smooth closed curve will be obtained as shown in Fig. 3b. The Bézier curves obtained from the control points in Fig. 3a and 3b are also shown for comparison.

To design a free-form shape by a Bézier$^+$ curve, the designer selects a sequence of control points just like in a Bézier curve. Initially, tangent directions are estimated from the control points using formulas (12) and (13). The magnitude of the tangents is initially set to a fixed value such as 1. Parameter s is then interactively revised with the motion of the mouse or press of a key until the curve gets sufficiently close to the control points. The shape of the curve is then locally modified by pointing somewhere along the control line and rotating the line by a desired amount about the associating control point.

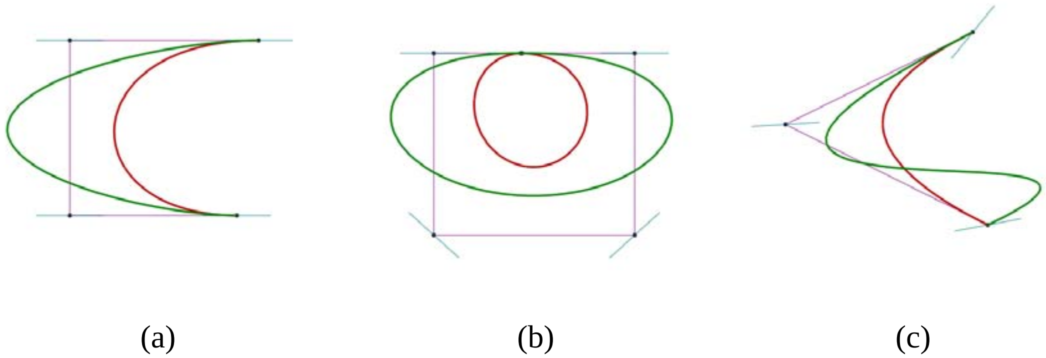


Figure 3: Comparison of Bézier and Bézier$^+$ curves. (a), (b) Both curves interpolate the first and last control points. The interior tangent directions in Bézier$^+$ are found such that tangents at end points become the same as tangents provided by the end control lines. (c) A Bézier$^+$ curve may contain one inflection point more than the Bézier curve obtained from the same control points.

3 Properties of Bézier$^+$ Curves

A Bézier$^+$ curve obtained from n control points will be of degree n. Therefore, Bézier$^+$ curves may contain one inflection point more than the Bézier curve obtained from the same set of control points. This implies that a Bézier$^+$ curve of order n intersects a line at most at n points. Note that a Bézier curve of order n intersects a line at most at $n-1$ points. An example comparing Bézier and Bézier$^+$ curves obtained from 3 control points is shown in Fig. 3c.

If all control lines of a Bézier$^+$ curve lie on a line, the obtained Bézier$^+$ curve will lie on the line also. This is because the curve is a convex combination of its control lines (see equation (3) or (5)).

Bézier$^+$ curves are affine invariant if the affine transformation is applied to control lines of the curve. This is immediately obvious from equation (3) or (5). Transforming the control lines of a Bézier$^+$ curve by an affine transformation will be the same as transforming the Bézier$^+$ curve, point by point, by the same affine transformation.

Equations (10) and (11) can also be written as follows:

$$x^+(u) = \sum_{i=0}^{k} \left[(1-u)x_i^1 + ux_i^2\right] B_{i,k}(u), \tag{14}$$

$$y^+(u) = \sum_{i=0}^{k} \left[(1-u)y_i^1 + uy_i^2\right] B_{i,k}(u) \tag{15}$$

where

$$x_i^1 = x_i - sX_i^{'}u_i, \qquad y_i^1 = y_i - sY_i^{'}u_i, \qquad i = 0, \ldots, k, \tag{16}$$

$$x_i^2 = x_i + sX_i^{'}(1-u_i), \quad y_i^2 = y_i + sY_i^{'}(1-u_i), \quad i = 0, \ldots, k. \tag{17}$$

Equations (14) and (15) define a Bézier$^+$ curve as a linear combination of two Bézier curves with control points given in (16) and (17).

Since each of the Bézier curves defined in a Bézier$^+$ curve provide the convex-hull property, the curve obtained from the linear combination of the two Bézier curves will fall inside the convex hull of the control points for both curves. Therefore, to determine the convex hull of the Bézier$^+$ curve, from the control lines, the control points defined by (16) and (17) are determined and the convex hull of the combined control points is computed.

Consider designing a curve segment from four control points as shown in Fig. 4. If the Bézier formulation is used, the interior curve is obtained, while if the Bézier$^+$ formulation is used, the exterior curve is obtained. Note that the Bézier$^+$ curve is much closer to the selected points than the Bézier curve. When a user selects control points like these, the user anticipates a control polygon that roughly approximates the shape. Having a mental picture of a desired shape, the user selects a number of points on the shape in order to obtain a curve that roughly resembles the control polygon. A Bézier curve is not only quite far away from its interior control points, the interior control points are points on the control polygon that are locally farthest from the curve. To create a Bézier curve similar to the Bézier$^+$ curve shown in Fig. 4, it is necessary to move the control points outside the viewing window, making interaction with the curve impossible. Alternatively, the curve can be subdivided into small segments and each segment can be designed by a Bézier curve. This, however, requires considerable user interaction. In interactive design, it is desirable to create a shape with as few curve segments and in as little time as possible.

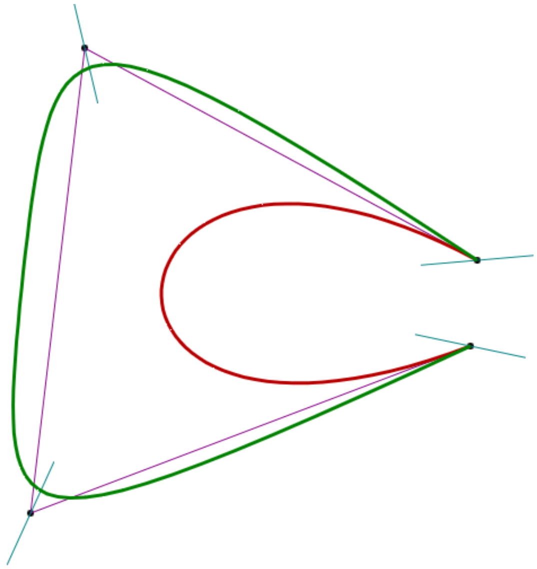


Figure 4: Comparison of Bézier and Bézier$^+$ curves obtained from the same set of control points. The interior curve is the Bézier and the exterior curve is the Bézier$^+$.

Each component of a Bézier curve can be considered a weighted sum of constant functions. For instance, $x(u)$ is defined in terms of the x-component of the control points. The x-component of control point i (x_i) can be considered line $x = x_i$ in the xu plane. The x component of a Bézier curve is, therefore, obtained from the weighted sum of lines parallel to the u axis in the xu plane. Since the slope of line $x = x_i$ is 0 and the slope of the ith basis function at $u = u_i$ is also 0, if a slope other than 0 is required at $u = u_i$ where $i \neq 0$ and $i \neq k$, the contribution of basis functions other than those centered at u_i need to be used to create the curve. This means that at $u = u_i$, the basis function associated with that parameter coordinate has no influence on the slope of the curve. To produce a desired slope at $u = u_i$, control points other than $\mathbf{V}_i$ should be moved. This makes creation of desired local shapes very difficult.

If we consider a component of a Bézier$^+$ curve, say $x^+(u)$, we see that the formulation still represents a weighted sum of lines, but the slopes of the lines may no longer be 0. The user can interactively specify the slopes of the lines, and the proposed formulation will use the specified slopes to create desired local shapes. The influence of control line $lx_i(u)$ is maximum at $u = u_i$, and its influence fades away as one moves away from u_i due to the monotonically decreasing nature of the basis functions. The ability of the Bézier$^+$ curve to revise a shape locally using the tangent directions makes it possible to create and revise a free-form shape more effectively than the Bézier curve.

Attempts to define curves with lines have been made before. Hoschek [10] introduced the idea of dual curves, treating Bézier control points as lines in the homogeneous coordinate system to create duals of Bézier curves. The idea introduced here is not the same as the dual curve idea. A curve and its dual do not look alike and have very different properties. For example, an inflection point in one necessitates a cusp in the other. Curves obtained by control points and control lines as described here are very similar and have similar properties.

There is similarity between control-line curves and Catmull-Rom splines [2]. In the Catmull-Rom formula, a curve is defined by a blending of piecewise polynomials. When the polynomial segments are of degree 1, the Catmull-Rom spline and the control-line curve become similar. Although tangent magnitudes in Catmull-Rom splines are fixed. In the proposed formula, tangent magnitudes may be varied to create different curves from the same set of lines.

In the preceding sections, control-line curves were defined using Bernstein-Bézier basis functions. Similarly, control-line curves can be defined using B-splines [7, 11], rational Bézier [11], non-uniform rational B-spline (NURBS) [6], and rational Gaussian (RaG) basis functions [8]. The control-line idea can be extended to surfaces also. Next, we will show that if the control points are replaced with control planes in a parametric surface, a formulation will be obtained that enables the design of free-form shapes by modifying the control-plane normals. This idea will be demonstrated in the context of Bézier surfaces, but the idea is general and can be applied to B-spline, rational Bézier, NURBS, RaG, and other parametric surfaces that are defined in terms of control points.

4 Bézier and Bézier$^+$ Surfaces

A Bézier surface obtained from a grid of $(m+1) \times (n+1)$ control points is defined by [11]

$$\mathbf{P}(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{V}_{ij} B_{i,m}(u) B_{j,n}(v), \tag{18}$$

where $B_{i,m}$ is the ith Bernstein polynomial of degree m, $B_{j,n}$ is the jth Bernstein polynomial of degree n, and $\mathbf{V}_{ij}$ is the ijth control point. We define a Bézier$^+$ surface by replacing the control points with control planes:

$$\mathbf{P}^+(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{L}_{ij}(u,v) B_{i,m}(u) B_{j,n}(v), \tag{19}$$

where $\mathbf{L}_{ij}$ is defined by

$$\mathbf{L}_{ij}(u,v) = \mathbf{p}^u(u_{ij},v_{ij})(u-u_{ij}) + \mathbf{p}^v(u_{ij},v_{ij})(v-v_{ij}) + \mathbf{l}_{ij}. \tag{20}$$

$\mathbf{l}_{ij}$ is found such that $\mathbf{L}_{ij}(u_{ij},v_{ij}) = \mathbf{V}_{ij}$. Therefore,

$$\mathbf{L}_{ij}(u,v) = \mathbf{V}_{ij} + \mathbf{p}^u(u_{ij},v_{ij})(u-u_{ij}) + \mathbf{p}^v(u_{ij},v_{ij})(v-v_{ij}), \tag{21}$$

and so

$$\begin{aligned}\mathbf{P}^+(u,v) &= \sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{V}_{ij}B_{i,m}(u)B_{j,n}(v)\\ &+ s\sum_{i=0}^{m}\sum_{j=0}^{n}\Big[\mathbf{P}^u(u_{ij},v_{ij})(u-u_{ij}) + \mathbf{P}^v(u_{ij},v_{ij})(v-v_{ij})\Big]B_{i,m}(u)B_{j,n}(v). \end{aligned}\tag{22}$$

s represents the magnitude of all control-plane normals and $\mathbf{P}^u(u_{ij},v_{ij})$ and $\mathbf{P}^v(u_{ij},v_{ij})$ are the gradients of the ijth control plane with respect to u and v. The first term in equation (22) is a regular Bézier surface. A Bézier$^+$ surface is, therefore, defined in terms of a Bézier surface and an adjustment to it defined by the control-plane normals. Although a single Bézier surface is obtained from a grid of control points, with the Bézier$^+$ formulation a variety of surfaces can be generated from the same grid of control points.

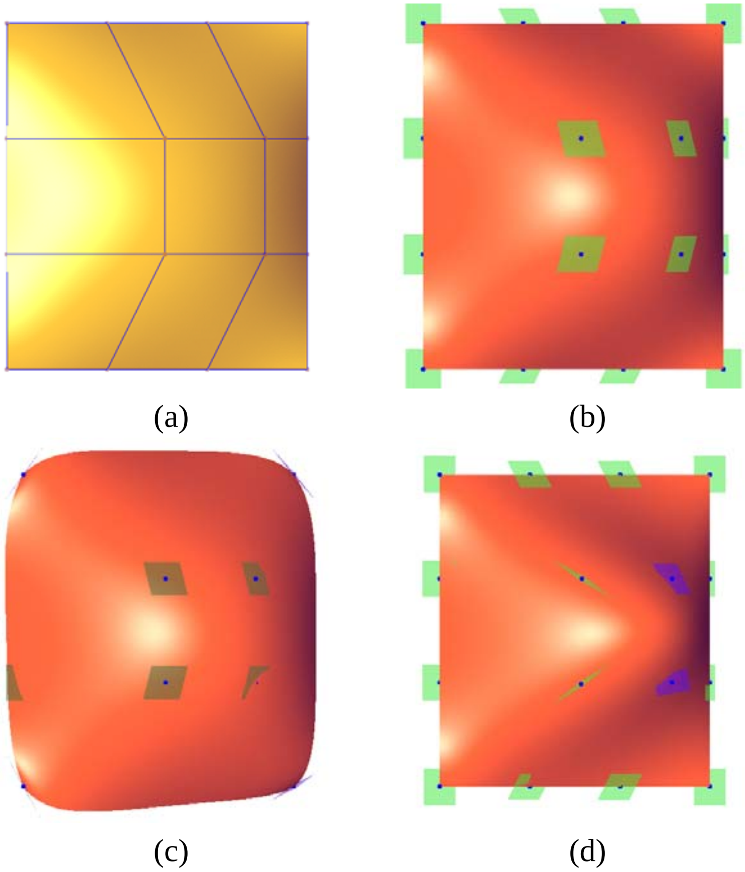
To create a Bézier$^+$ surface interactively, initially $\mathbf{P}^u(u,v)$ and $\mathbf{P}^v(u,v)$ are estimated from the control points:

$$\mathbf{P}^u(u_{ij},v_{ij}) = \frac{\mathbf{V}_{i+1,j}-\mathbf{V}_{i-1,j}}{||\mathbf{V}_{i+1,j}-\mathbf{V}_{i-1,j}||}, \tag{23}$$

$$\mathbf{P}^v(u_{ij},v_{ij}) = \frac{\mathbf{V}_{i,j+1}-\mathbf{V}_{i,j-1}}{||\mathbf{V}_{i,j+1}-\mathbf{V}_{i,j-1}||}, \tag{24}$$

For the boundary control points, $\mathbf{V}_{-1,j}$ is replaced with $\mathbf{V}_{0,j}$, $\mathbf{V}_{m+1,j}$ is replaced with $\mathbf{V}_{m,j}$, $\mathbf{V}_{i,-1}$ is replaced with $\mathbf{V}_{i,0}$, and $\mathbf{V}_{i,n+1}$ is replaced with $\mathbf{V}_{i,n}$ in equations (23) and (24). Once the initial surface is drawn, the user can vary the control-plane normals individually to create the desired local shapes. $\mathbf{P}^u(u_{ij},v_{ij}) \times \mathbf{P}^v(u_{ij},v_{ij})$ is the normal vector associated with control point $\mathbf{V}_{ij}$. We draw a planar segment centered at each control point and allow the user to interactively change the orientation of the plane or the normal to it using the motion of the mouse. This will revise the shape of the surface in the neighborhood of the control point.

An example comparing Bézier and Bézier$^+$ surfaces is shown in Fig. 5. The Bézier surface obtained from a grid of 4×4 control points is shown in Fig. 5a. The Bézier$^+$ surfaces obtained from the same control points are shown in Figs. 5b–d. The control planes in Fig. 5b are those initially computed from (23) and (24). Fig. 5c shows the result of modifying the control-plane normals at the four corners of the surface, and Fig. 5d shows the surface obtained by varying the control-plane normals at the four interior control points. A variety of surface shapes can be created in this manner from a single grid of control points as opposed to the Bézier representation, which produces a single surface from a given grid of control points. To obtain surfaces similar to those shown in Figs. 5b–d by the Bézier formulation, it is required to move some of the control point, outside the viewing volume, making interaction with the surface impossible. Alternatively, the surface may be subdivided into small segments, and each segment may be designed separately by a Bézier patch, but this will considerably increase user interaction.


(a) (b)

(c) (d)

Figure 5: (a) Bézier and (b)–(d) Bézier$^+$ surfaces obtained from the same control points but different control-plane normals.

5 Properties of Bézier$^+$ Surfaces

Properties of Bézier$^+$ surfaces follow those of Bézier$^+$ curves. When magnitude of normals at the control points approach zero, a Bézier$^+$ surface approaches the Bézier surface obtained by the control points. As the magnitude of normals increase, the surface expands more. This makes it possible to generate a surface that passes very close to its control points.

Each component of a Bézier surface can be considered a weighted sum of horizontal planes in the uv space. This makes the creation of desired local tangents in a Bézier surface very difficult. Each component of a Bézier$^+$ surface, on the other hand, is defined by a weighted sum of planes with desired slopes in the parameter space, making the creation of required local shapes much easier.

A Bézier$^+$ surface can be considered a Catmull-Rom spline [2] if the given tangent planes are blended using Bernstein polynomials. Although a Catmull-Rom spline produces a single surface from a given set of planes, the proposed formulation can create a variety of surfaces by varying the magnitude of normals to the planes.

Analogous to Bézier$^+$ curves, Bézier$^+$ surfaces can be written as a linear combination of two

Bézier surfaces, where the control points of each Bézier surface are defined in terms of the Bézier$^+$ control points and associating normals. Therefore, a Bézier$^+$ surface falls inside the convex hull of the control points of the two Bézier surfaces defining it.

A Bézier$^+$ surface passes through the four-corner control points of its control grid. This is because the second term in equation (22) becomes zero at $u = 0, v = 0; u = 0, v = 1; u = 1, v = 0$; and $u = 1, v = 1$. A Bézier$^+$ surface, in general, is not tangent to its control mesh at the four corners, but the surface can be made to become tangent to it by adjusting the control-plane normals at four interior control points by solving a system of equations.

Bézier$^+$ surfaces are affine invariant. That is, transforming the control planes of a Bézier$^+$ surface with an affine transformation, we will obtain a surface that is the same as the surface obtained by transforming the original surface point by point using the affine transformation. This is immediately obvious from equation (22). If all the control planes fall in a plane, the obtained Bézier$^+$ surface will also fall in the plane. This is because a Bézier$^+$ surface is obtained from a convex combination of its control planes.

6 Representation of Irregular Meshes by RaG$^+$ Surfaces

The design of some geometric models requires the use of irregular control meshes. Triangulation of range data and simplification of triangle meshes often produce irregular meshes with non-uniform elements. Subdivision techniques may be used to approximate the meshes with smooth surfaces [13, 14, 15]. Alternatively, triangular, quadrilateral, and in general n-sided patches may be used to create a piecewise smooth approximation to a polygon mesh [3, 4, 9, 12]. In the following, it will be shown that if polygon faces are taken as the control planes in the equation of control-plane rational Gaussian (RaG$^+$) surfaces, an efficient means to represent polygon meshes with smooth parametric surfaces will be obtained. The smoothness parameter of a RaG$^+$ surface may be varied to create surfaces at different smoothness or resolution levels.

Suppose a given polygon mesh has N faces, and $\mathbf{V}_i$ is the center of the ith polygon face obtained by averaging the vertices of that face. A RaG surface approximating $\{\mathbf{V}_i : i = 1, \ldots, N\}$ is written by [8]:

$$\mathbf{P}(u, v) = \sum_{i=1}^{N} \mathbf{V}_i g_i(u, v), \qquad u, v \in [0, 1], \tag{25}$$

where $g_i(u, v)$ is the ith basis function of the surface defined by

$$g_i(u, v) = \frac{G_i(u, v)}{\sum_{j=1}^{N} G_j(u, v)}, \tag{26}$$

and $G_i(u, v)$ is a 2-D Gaussian of height 1 centered at (u_i, v_i):

$$G_i(u, v) = \exp\{-[(u - u_i)^2 + (v - v_i)^2]/2\sigma^2\}. \tag{27}$$

(u_i, v_i) are the parameter coordinates associated with $\mathbf{V}_i$, which is obtained from the average of parameter coordinates at the vertices of the ith polygon face.

Note that equation (25) is defined by a single sum, showing that a regular grid of control points is not required to define a RaG surface and an irregular grid of control points is sufficient to obtain a parametric surface. Now, if the control points of a RaG surface are replaced with the control planes defined by the polygon faces, the obtained surface will approximate the polygon mesh.

Replacing control points $\{\mathbf{V}_i : i = 1, \ldots, N\}$ in equation (25) with control planes $\{\mathbf{l}_i(u, v) : i = 1, \ldots, N\}$, we will have

$$\mathbf{P}^{+}(u, v) = \sum_{i=1}^{N} \mathbf{l}_i(u, v) g_i(u, v), \qquad u, v \in [0, 1]. \tag{28}$$

$\mathbf{l}_i(u, v)$ is determined by making plane

$$\mathbf{l}_i(u, v) = \mathbf{a}u + \mathbf{b}v + \mathbf{c} \tag{29}$$

pass through the center and two of the vertices of the ith polygon face in the parameter space. When $u = u_i$ and $v = v_i$, $\mathbf{l}_i(u_i, v_i) = \mathbf{V}_i$. Therefore, the formulation for a RaG$^+$ surface becomes:

$$\mathbf{P}^{+}(u, v) = \sum_{j=0}^{N} \mathbf{V}_i g_i(u, v) + \sum_{j=0}^{N} \Big[\mathbf{a}(u - u_i) + \mathbf{b}(v - v_i)\Big] g_i(u, v). \tag{30}$$

$\mathbf{a}$ and $\mathbf{b}$ are gradients of the ith control plane with respect to u and v. These gradients are determined by fitting a parametric plane as defined by (29) to the center and two vertices of a polygon face, finding the partial derivative of the plane with respect to u and v, and evaluating the partial derivatives at (u_i, v_i).

The standard deviation of Gaussians, σ, may be used to control the resolution or degree of smoothness/detailedness of an obtained surface. As σ is decreased, more details are reproduced, and as σ is increased, less details are reproduced, creating a smoother surface.

Examples of RaG$^+$ surfaces approximating large polygon meshes are shown in Fig. 6. Fig. 6a shows a digital elevation map covering an area over the Grand Canyon. This data set is courtesy of the U.S. Geological Survey. The figure shows the triangulation and rendering of the elevation data in shaded form. Approximating the mesh by a RaG$^+$ surface at two different smoothness levels, we obtain Figs. 6b and 6c. When σ is very small, the surface follows the mesh closely and reproduces the noisy details in data. As σ is increased, noisy details reduce, capturing more of the global geometry of the scene.

Fig. 6d shows the triangulated Stanford Bunny in shaded form. Approximating the mesh with a RaG$^+$ surface at two smoothness levels, we obtain the surfaces depicted in Figs. 6e and 6f. Again, as the smoothness parameter is increased, more local details are lost, creating a smoother surface.

Fig. 6g shows a triangulated and shaded rendering of a range scan of a person's knee. This data set is courtesy of SGI. Approximation of this triangle mesh by RaG$^+$ surfaces at two different smoothness levels are shown in Figs. 6h and 6i. This representation allows a user to reproduce a desired amount of local details in a shape.

RaG$^+$ is a global formulation and theoretically a surface point depends on all the control planes. In practice, however, since Gaussians decay exponentially, only control planes associated with control points within a small neighborhood of (u, v) in the parameter space will affect surface point $\mathbf{P}^{+}(u, v)$. The neighborhood size depends on parameter σ and the required accuracy of the computation. The σ used to obtain Figs. 6b, 6e, and 6h was 0.002. Control planes centered at parameter coordinates farther than 5σ from (u, v) in the parameter space will not have any effect on the displayed surface because of the digital nature of display devices. Therefore, it is sufficient to consider control planes centered at parameter coordinates within $[u-0.01, u+0.01] \times [v-0.01, v+0.01]$ to calculate $\mathbf{P}^+(u,v)$. Therefore, only 4/10000th of all control planes were needed to find a surface point in Figs. 6b, 6e, and 6h. To provide quick access to the relevant control planes during computation of a surface, the control planes are initially binned in a 2-D array based on the parameter coordinates of the associating control points. Then, to find a surface point at (u,v), a table look up is performed to find those control planes that affect the surface point and use them in the calculations. Therefore, although the formulation is global, efficient algorithms can be devised to quickly select with required accuracy and smoothness the control planes that affect a surface point, using only the selected control planes in the calculations.

7 Conclusions

New formulations for parametric curves and surfaces were presented that use control lines and control planes instead of control points. These formulations provide more flexibility in design than traditional formulations because, in addition to the control points, tangents at the control points are used to design 2-D and 3-D shapes. The tangents provide more local shape control in geometric modeling.

The control-line/control-plane idea was demonstrated in the context of Bézier curves and surfaces, but the idea is general and can be applied to B-spline, NURBS, rational Bézier, and in general, all formulations that use control points. If the idea is applied to a RaG surface, since the control points of a RaG surface can be irregularly spaced, a RaG$^+$ surface can approximate an irregular polygon mesh.

References

[1] P. Bézier, *Numerical Control-Mathematics and Applications* (translated by A. R. Forrest), John Wiley and Sons, London, 1972.

[2] E. Catmull and R. Rom, A class of local interpolating splines, *Computer Aided Geometric Design*, Barnhill and Riesenfeld (eds.), Academic Press, 1974, pp. 317–326.

[3] C. K. Chui and M.-J. Lai, Filling polygonal holes using C^1 cubic triangular spline patches, *Computer Aided Geometric Design*, vol. 17, 2000, pp. 297–307.

[4] P. Constantini and C. Manni, A local shape-preserving interpolation scheme for scattered data, *Computer Aided Geometric Design*, vol. 16, 1999, pp. 385–405.

[5] G. Farin, Surfaces over Dirichlet tesselation, *Computer Aided Geometric Design*, vol. 7, 1990, pp. 281–292.

[6] G. Farin, *NURBS: From Projective Geometry to Practical Use*, Second Edition, A. K. Peters, 1999.

[7] W. J. Gordon and R. Riesenfeld, B-spline curves and surfaces, *Computer Aided Geometric Design*, Barnhill and Riesenfeld (eds.), Academic Press, 1974, pp. 95–126.

[8] A. Goshtasby, Geometric modeling using rational Gaussian curves and surfaces, *Computer Aided Design*, vol. 27, no. 5, 1995, pp. 363–375.

[9] J. C. Navau and N. P. Garcia, Modeling surfaces from meshes of arbitrary topology, *Computer Aided Geometric Design*, vol. 8, 1991, pp. 89–96.

[10] J. Hoschek, Dual Bézier curves and surfaces, *Surfaces in CAGD*, R. E. Barnhill and W. Boehm (eds.), North-Holland Publishing, 1983, pp. 147–156.

[11] M. E. Mortenson, *Geometric Modeling*, Second Edition, Wiley Computer Publishing, 1997.

[12] L. A. Shirman and C. H. Sequin, Local surface interpolation with shape parameters between adjoining Gregory patches, *Computer Aided Geometric Design*, vol. 7, 1990, pp. 375–388.

[13] J. Stam, On subdivision schemes generalizing uniform B-spline surfaces of arbitrary degree, *Computer Aided Geometric Design*, vol. 18, 2001 pp. 383–396.

[14] L. Velho and Denis Zorin, 4–8 subdivision, vol. 18, 2001, pp. 397–427.

[15] D. Zorin and P. Schröder, A unified framework for primal/dual quadrilateral subdivision schemes, *Computer Aided Geometric Design*, vol. 18, 2001, pp. 429–454.

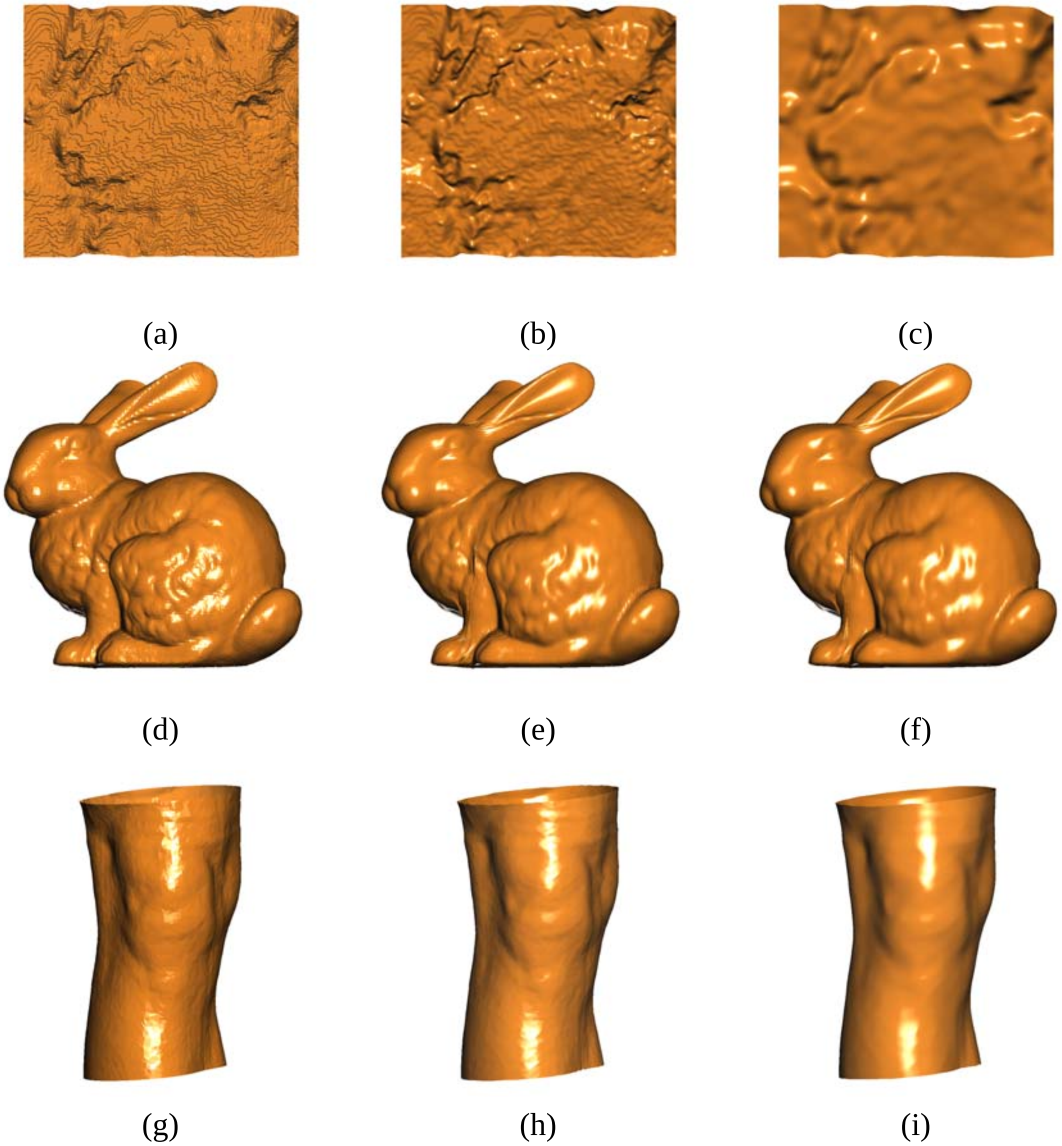


Figure 6: (a) A digital elevation map covering an area over the Grand Canyon after triangulation and rendering in shaded form. (b), (c) RaG$^+$ surfaces approximating the triangle mesh at two resolutions. (d) The triangulated Stanford Bunny shown in shaded form. (e), (f) Approximation of the Stanford Bunny by RaG$^+$ surfaces at two resolutions. (g) A range scan of a person's knee after triangulation and rendering in shaded form. (h), (i) Approximation of the triangulated knee by RaG$^+$ surfaces at two resolutions.