Generalized B-spline Surfaces of Arbitrary Topology

Charles Loop and Tony DeRose
University of Washington

Abstract

B-spline surfaces, although widely used, are incapable of describing surfaces of arbitrary topology. It is not possible to model a general closed surface or a surface with handles as a single non-degenerate B-spline. In practice such surfaces are often needed. In this paper, we present generalizations of biquadratic and bicubic B-spline surfaces that are capable of capturing surfaces of arbitrary topology (although restrictions are placed on the connectivity of the control mesh). These results are obtained by relaxing the sufficient but not necessary smoothness constraints imposed by B-splines and through the use of an n-sided generalization of Bézier surfaces called S-patches.

CR Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling - curve, surface, solid, and object representations; J.6 [Computer-Aided Engineering]: Computer-Aided Design (CAD); G.1.2 [Approximation]: Spline Approximation.

Additional Key Words and Phrases: Computer-aided geometric design, B-spline surfaces, n-sided patches, geometric continuity.

1 Introduction

Parametric surfaces have proven themselves an excellent tool for representing smoothly varying sculptured objects. B-splines have emerged as the polynomial basis of choice for working with parametric surfaces. However, the current theory of B-splines has serious shortcomings when modeling general closed surfaces or surfaces with handles.

A B-spline surface is a deformation of a planar domain, tessellated into a regular grid of rectangles. It is quite natural for the surface to be treated as a collection of tensor product polynomial patches defined over these rectangles. This leads to notions of *parametric continuity* (denoted C^k continuity), where smoothness is defined in terms of matching derivatives along patch boundaries. It is precisely this treatment that limits the possible topologies of a B-spline surface.

A more general view considers the spline surface to be a collection of (possibly rational) polynomial maps from independent n-sided polygonal domains, whose union possesses continuity of some number of geometric invariants, such as tangent planes. In this view, patches are required to meet with *geometric continuity* (denoted G^k continuity), a measure of continuity that subsumes strict parametric continuity. This more general view allows patches to be sewn together to describe free form surfaces in much richer and more complex ways.

The n-sided element we use is the S-patch developed in [18]. Using the S-patch form, our surfaces possess many desirable properties. One of the most important of these properties is that S-patches generalize both Bézier tensor product and triangular patches, meaning that an S-patch based geometric modeler is compatible with existing popular patch types. Since all S-patches are instances of a single general structure, algorithms may be derived that are independent of the number of sides n, leading to uniformity and simplicity. This dispels the notion that inclusion of n-sided patches into a geometric modeler would increase complexity due to an increase in special cases. The S-patch form is more complicated than tensor product forms. However, it is also very structured, so we believe the increase in generality offsets the increase in complexity.

By relaxing C^1 continuity to G^1 continuity, and by allowing n-sided S-patch elements, we obtain more general spline methods that contain B-spline surfaces as a proper subset. Just as for B-splines, our more general surfaces are created as smooth approximations to *control meshes*. By a control mesh we mean a collection of *control points*, or, synonymously, *control vertices*, together with connectivity information used to define edges and faces.[†] Here we offer two methods for transforming control meshes into G^1 spline surfaces. The methods differ in the connectivity restrictions placed on the control mesh. The first method is a generalization of biquadratic B-splines; it requires the control mesh to be constructed entirely from four sided faces, although any number of faces may meet at a vertex. The second method is a generalization of bicubic B-splines; it requires that exactly four faces meet at a vertex, although faces may contain any number of edges.

There are two reasons for the choice of restrictions on the control meshes. First, the restrictions are sufficiently relaxed to describe surfaces of arbitrary topology. Second, the restrictions are sufficiently strong to guarantee that all our surfaces have exactly four patches meeting at each interior corner. This specialization is exploited to yield a simple solution to the "twist compatibility" problem, a system of constraints that arises at interior corners. These schemes, therefore, retain the

This work was supported in part by the Xerox Corporation, IBM, Hewlett-Packard, the Digital Equipment Corporation, and the National Science Foundation under grants CCR-8957323 and DMC-8802949.

Authors' address: Department of Computer Science and Engineering, FR-35, University of Washington, Seattle, WA 98195.

[†] A control mesh is more precisely defined as a subdivision of a topological 2-manifold, possibly with boundary [12].

simple design features while overcoming the severe topological restrictions of traditional B-spline methods. It should be emphasized that these schemes are generalizations of *uniform* B-splines, although the extensions to the non-uniform and rational cases (NURBs) pose no fundamental difficulties.

1.1 Overview

After a review of previous work, given in Section 2, and a review of the basic properties of S-patches, given in Section 3, the principal results of this work are presented in a bottom up fashion. In Section 4, we solve the *n-sided hole problem* using S-patches. In this problem the existence of an n-sided hole surrounded by polynomial patches is assumed. The objective is to find a single S-patch that meets the surrounding patches with G^1 continuity. In Section 5, we use the solution of the n-side hole problem to define a patch representation that mimics the relatively simple continuity requirements of Bézier tensor product patches. Hence these patches are easy to join together with G^1 continuity. In section 6 we presents the two generalized B-spline schemes for modeling surfaces with arbitrary topology. Like B-splines, both of these schemes produce surfaces that are *smoothed* versions of control meshes. Unlike B-splines, however, our control meshes are capable of modeling surfaces of arbitrary topology. Finally, in Section 7, we offer some concluding remarks and directions for future work.

2 Previous Work

When modeling with B-spline surfaces, a common way to subvert the topological constraints is to introduce degeneracies into the control meshes. This amounts to collapsing one or more edges of the control mesh to a point, resulting in one or more 3-sided faces. This causes an irregularity in the parameterization, meaning that partial derivatives are not linearly independent. Degeneracies of this sort introduce various problems, such as calculating normal vectors. In an effort to find more robust solutions, a significant amount of recent work has been done in the areas of geometric continuity, non-tensor product patches, and generalizing B-splines.

It has been shown (c.f. Herron [15]) that it is impossible to construct closed, non-degenerate C^1 surfaces of arbitrary topology. One solution to this is to replace parametric continuity with geometric continuity, a topic that has received substantial study in recent years (cf. [3, 8, 16]).

To address the problem of irregular patch networks, many non-tensor product patches have been developed. These include the n-sided patches developed by Sabin [19], and Hosaka and Kimura [17], which are limited to at most 5 and 6 sides respectively. The patch described by Gregory [11] is not limited in the number of sides, and is similar in spirit to the Coons patch. Other true n-sided patch representations have been proposed by Herron [14], and by Varady [22]. The *Gregory Patches* of Chiyokura and Kimura [6] are generalizations of Bézier tensor product patches that contain (removable) singularities at patch corners.

Several generalizations of B-splines have been proposed. The earliest of these are the *recursive subdivision* schemes of Doo and Sabin [9], and Catmull and Clark [4]. These methods allow arbitrary control meshes and generally produce pleasing surfaces; however, the surfaces are defined as the limit of a local averaging procedure and do not, in general, possess a closed form parameterization. A generalization of B-splines that makes use of parametric surfaces has been found by van Wijk [21]. This scheme uses tensor product patches exclusively and imposes relatively strict requirements on the form of the control mesh.

3 S-patches

As mentioned in Section 1, the generalized B-spline schemes are based on S-patches. S-patches refer to a generalization of Bézier surfaces where any number n of boundary curves are permissible. The underlying theory has been established elsewhere [18]. In this section, we briefly summarize the results necessary to develop the generalized B-spline schemes.

As is shown in [18], S-patches possess a rich structure, largely because they are defined in terms of multivariate Bernstein polynomials and Bézier simplexes. It is therefore convenient to begin with a discussion of Bézier simplexes.

In what follows, multi-indices will be denoted by italic characters with a diacritical arrow, as in $\vec{i}$. Multi-indices are tuples of non-negative integers, the components of which are subscripted starting at one; for instance, $\vec{i} = (i_1, ..., i_{k+1})$. The norm of a multi-index $\vec{i}$, denoted by $|\vec{i}|$, is defined to be the sum of the components of $\vec{i}$. The symbol $\vec{e_j}$ denotes a multi-index whose components are all zero except for the jth component which is one. Addition, subtraction and scalar multiplication of multi-indices are defined componentwise.

By setting $\vec{i} = (i_1, ..., i_{k+1})$ and requiring $|\vec{i}| = d$, the k-variate Bernstein polynomials of degree d can be defined by

$$B^d_{\vec{i}}(u_1, ..., u_{k+1}) = \binom{d}{\vec{i}} u_1^{i_1} u_2^{i_2} \cdots u_{k+1}^{i_{k+1}}$$

where $\binom{d}{\vec{i}}$ is the multinomial coefficient defined by

$$\binom{d}{\vec{i}} = \frac{d!}{i_1!\, i_2! \cdots i_{k+1}!},$$

and where $u_1, ..., u_{k+1}$ are real numbers that sum to one. It is known [7] that for every polynomial $Q : X_1 \rightarrow X_2$ of degree d, where X_1 is an affine space of dimension k and X_2 is an affine space of arbitrary dimension, there exist unique points $\mathbf{V}_{\vec{i}} \in X_2$, $|\vec{i}| = d$, such that

$$Q(u) = \sum_{\vec{i}} \mathbf{V}_{\vec{i}} B^d_{\vec{i}}(u_1, ..., u_{k+1}), \qquad (1)$$

where $u_1, ..., u_{k+1}$ are the barycentric coordinates of $u \in X_1$ relative to a *simplex* $\Delta = \{v_1, ..., v_{k+1}\}$. (A k-simplex is a collection of $k + 1$ points such that none of the points can be written as an affine combination of the others. For example, the points of a 2-simplex form a triangle, and the points of a 3-simplex form a tetrahedron.)

Summations such as the one in Equation 1 are intended to be taken over all multi-indices whose norm matches the degree of the Bernstein polynomial. Thus, in Equation 1, the multi-index $\vec{i}$ is to take on all values such that $|\vec{i}| = d$. A polynomial Q, when expressed as in Equation 1, is called a *Bézier simplex*. The points $\mathbf{V}_{\vec{i}}$ are called the *control net*, and Δ is called the *domain simplex*.

S-patches build on the theory of Bézier simplexes as follows. An n-sided S-patch S is a mapping from a domain n-gon P and is conceptually constructed in two phases: first, $P = \{p_1, ..., p_n\}$ is embedded into an intermediate domain simplex $\Delta = \{v_1, ..., v_n\}$ contained in an affine space Y of dimension $n - 1$; next a Bézier simplex is created using Δ as its domain; finally, S is defined as the composition of the embedding and the Bézier simplex.. That is, if $L : P \rightarrow \Delta$ represents the embedding, and if $B : \Delta \rightarrow \Re^3$ is the Bézier simplex, then

$$S(p) = B \circ L(p), \qquad p \in P, \qquad (2)$$

as indicated in Figure 1.

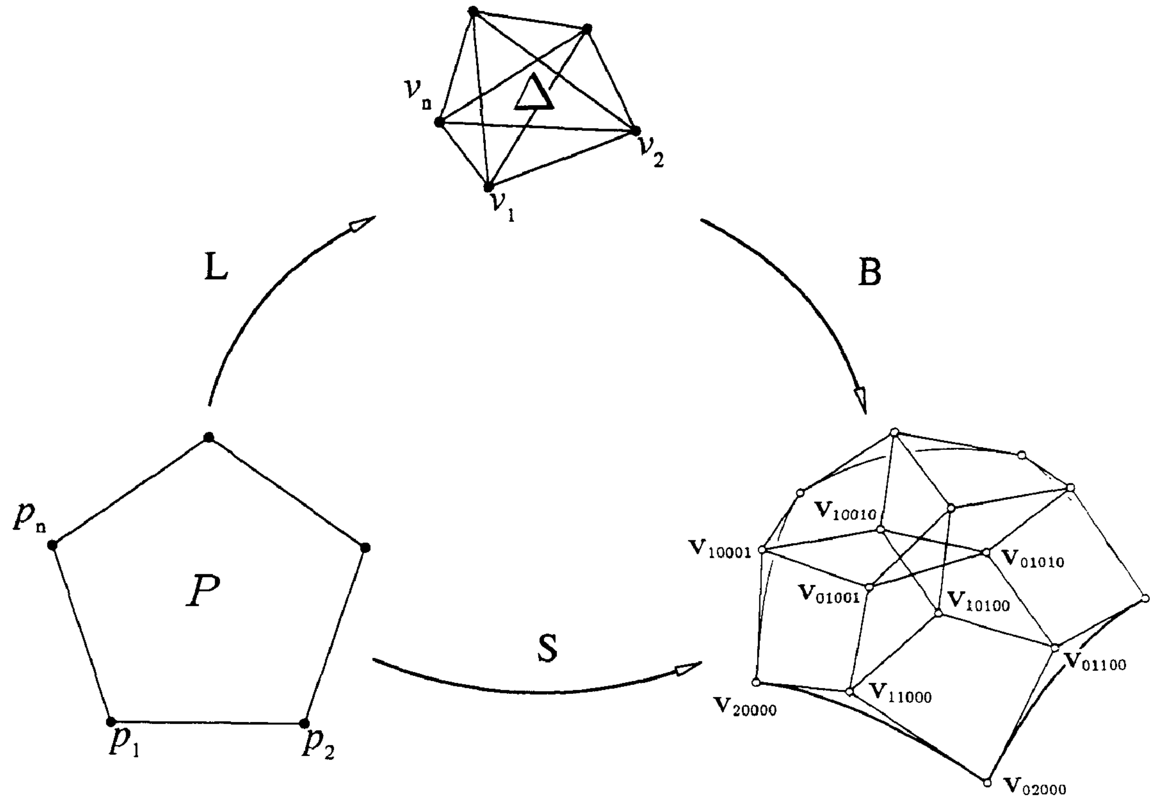

Figure 1: Schematic representation of S-patches.

To describe the embedding L we use to map the domain polygon P into the intermediate simplex Δ, we first introduce several helpful definitions. Let $\alpha_i(p)$ denote the signed area of the triangle pp_ip_{i+1}, where the sign is chosen to be positive if p is inside P. (Note: all indices are to be treated in cyclic fashion.) Let

$$\pi_i(p) = \alpha_1(p) \cdots \alpha_{i-2}(p) \cdot \alpha_{i+1}(p) \cdots \alpha_N(p),$$

for $i = 1, ..., n$, denote the product of all areas except for α_{i-1} and α_i, and let

$$\ell_i(p) = \frac{\pi_i(p)}{\pi_1(p) + \cdots + \pi_n(p)}.$$

With these definitions, every point $p \in P$ is mapped by L into the point

$$L(p) = \ell_1(p)v_1 + \ell_2(p)v_2 + \cdots + \ell_n(p)v_n, \qquad (3)$$

in Y. This embedding has the important property that it is *edge-preserving*, meaning that if p lies on an edge of P, then $L(p)$ lies on an edge of Δ; additionally, the interior of P is mapped into the interior of Δ [18].

If $\mathbf{V}_{\vec{i}}$ denotes the control net of a Bézier simplex B, then an S-patch S is defined as

$$S(p) = B \circ L(p) = \sum_{\vec{i}} \mathbf{V}_{\vec{i}} B^d_{\vec{i}}(\ell_1(p), ..., \ell_n(p)). \qquad (4)$$

REMARK : The definition of S-patches given in Equation 4 is a slight specialization of the definition given in [18], since in that work B was allowed to be a *rational* Bézier simplex. For brevity and simplicity, we have chosen not to introduce this additional complication here.

The integer d in Equation 4 is known as the *depth* of the S-patch, to avoid confusion with the polynomial degree of the patch, which is $d(n-2)$. The control net $\mathbf{V}_{\vec{i}}$ is taken as the control net of S, an example of which are shown in Figure 1. S-patch control nets consist of interconnected n-sided closed polygonal *panels*. For instance, in Figure 1 the points $\mathbf{V}_{20000}$, $\mathbf{V}_{11000}$, $\mathbf{V}_{10100}$, $\mathbf{V}_{10010}$, $\mathbf{V}_{10001}$ form one such panel.

The compositional structure of the S-patch S together with the edge-preserving character of the embedding L endows the S-patch representation with a number of useful properties, including (see [18] for proofs):

- Points on S can be computed using the multivariate version of deCasteljau's algorithm.
- Boundary curves are in Bézier form, implying that boundary curves can be individually controlled. For instance, referring to Figure 1, the boundary curve corresponding to the bottom boundary of the control net is a quadratic Bézier curve whose control points are $\mathbf{V}_{20000}, \mathbf{V}_{11000}, \mathbf{V}_{02000}$. Control points such as these are called ***boundary points.***
- The panels of a control net that contain boundary points are termed ***boundary panels***. The tangent plane variation along a boundary curve is determined entirely by the corresponding boundary panels. Referring again to Figure 1 for example, the tangent plane along the bottom boundary of the control net is entirely determined by the two panels $\mathbf{V}_{20000}$, $\mathbf{V}_{11000}$, $\mathbf{V}_{10100}$, $\mathbf{V}_{10010}$, $\mathbf{V}_{10001}$ and $\mathbf{V}_{11000}$, $\mathbf{V}_{02000}$, $\mathbf{V}_{01100}$, $\mathbf{V}_{01010}$, $\mathbf{V}_{01001}$.

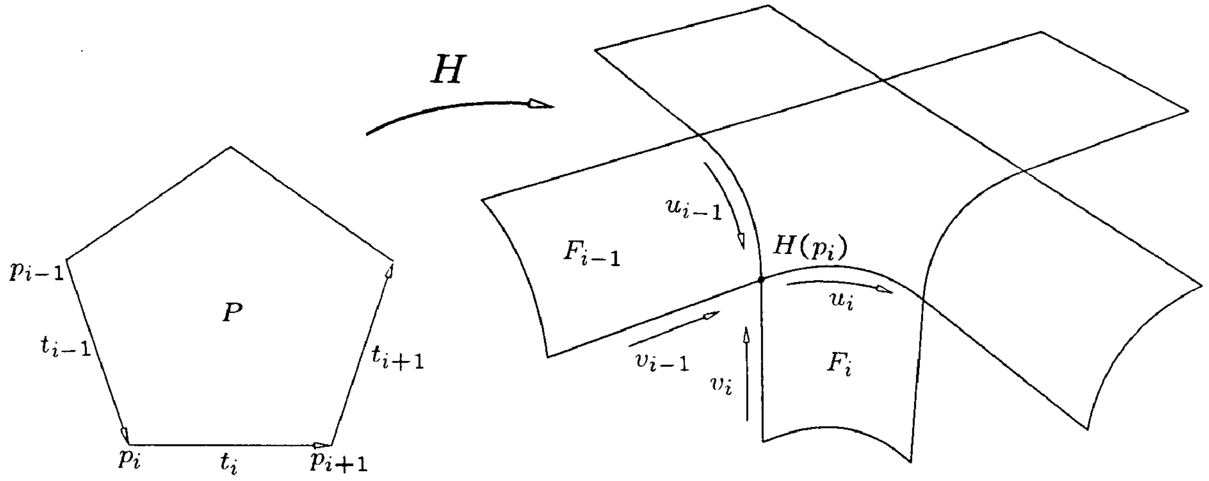

Figure 2: Patches $F_1, ..., F_n$ surrounding a n-sided hole.

- Given an S-patch control net of depth d describing a patch $S = B \circ L$, the S-patch control net of depth $d + 1$ for S can be constructed by executing the Bézier simplex degree raising algorithm on the control net of B. Thus, S-patch control nets can be ***depth elevated***.

- If P is a regular n-gon, and if Q is a polynomial patch in triangular Bézier form defined over some domain triangle T, then there is a simple algorithm for representing Q in S-patch form over P, referred to as the ***polynomial representation algorithm***.

- When $n = 3$, the S-patch form reduces to the standard Bézier triangle, and when $n = 4$, S-patches coincide with Bézier tensor product patches.

4 The N-sided Hole Problem

The n-sided hole problem arises in situations such as the one shown in Figure 2 where polynomial patches surround an n-sided hole. The objective is to construct a surface patch that fills the hole and meets the surrounding surfaces with at least G^1 continuity. Although this problem has been identified as an interesting problem in its own right [5, 10, 13], our reason for introducing and solving it here is two-fold: first, our solution fills the hole with a single S-patch; second, in Section 5 the solution is used to define a patch representation that is particularly convenient for developing the generalized B-spline schemes.

Referring to Figure 2, the hole to be filled is assumed to be surrounded by n patches $F_1, ..., F_n$, typically given in tensor product form; we wish to construct a single S-patch H that matches the surrounding patches with positional and tangent plane continuity. We shall find it convenient to introduce the following notation:

- The boundaries of the hole are given by $F_i(u_i, 1)$, $i = 1, ..., n$, as indicated by Figure 2.

- The domain polygon P of H is a regular n-gon with vertices $p_1, ..., p_n$. (Let $\vec{t}_i$ be the vector from p_i to p_{i+1}, and let E_i denote the i^{th} edge of P, that is, $E_i(t_i) = (1 - t_i)p_i + t_i p_{i+1}$, $t_i \in [0, 1]$.)

We assume that the patches surrounding the hole satisfy the following "twist compatibility" conditions:

$$\text{(A0)} \qquad F_{i-1}(1,1) = F_i(0,1)$$

$$\text{(A1)} \qquad \frac{\partial F_{i-1}}{\partial u_{i-1}}(1,1) = -\frac{\partial F_i}{\partial v_i}(0,1)$$

$$\text{(A2)} \qquad \frac{\partial F_{i-1}}{\partial v_{i-1}}(1,1) = \frac{\partial F_i}{\partial u_i}(0,1)$$

$$\text{(A3)} \qquad \frac{\partial^2 F_{i-1}}{\partial u_{i-1}\,\partial v_{i-1}}(1,1) = -\frac{\partial^2 F_i}{\partial u_i\,\partial v_i}(0,1)$$

We additionally assume that the patches are regular in the sense that partial derivatives are everywhere linearly independent.

Positional continuity of H with the surrounding surfaces is simply achieved by requiring that $H(E_i(t_i)) = F_i(t_i, 1)$, $t_i \in [0, 1]$, for each $i = 1, ..., n$. Differentiating this with respect to t_i implies that

$$\mathbf{D}_{\vec{t}_i} H(E_i(t_i)) = \frac{\partial F_i}{\partial u_i}(t_i, 1), \qquad t_i \in [0,1], \tag{5}$$

where $\mathbf{D}_{\vec{v}} f(p)$ denotes the directional derivative of f in the direction $\vec{v}$ at the point p.

We also require that H meets the surrounding surfaces with G^1 continuity. Along the i^{th} edge E_i this is equivalent to the existence of functions $\mu, \nu : [0, 1] \rightarrow \Re$ such that

$$\mathbf{D}_{-\vec{t}_{i-1}} H(E_i(t_i)) = \mu(t_i)\frac{\partial F_i}{\partial u_i}(t_i, 1) + \nu(t_i)\frac{\partial F_i}{\partial v_i}(t_i, 1), \tag{6}$$

for all $t_i \in [0, 1]$. This equation states that at each point on E_i, a cross boundary derivative of H, in this case taken in the direction of $-\vec{t}_{i-1}$, should be in the tangent plane of F_i, and hence should be expressible as a linear combination of F_i's first order partial derivative vectors. Conditions 5 and 6 are therefore sufficient (and in fact necessary) to guarantee that H and F_i share a common tangent plane along their common boundary curve (cf. Herron [16]).

The principal difficulty in constructing H is in determining the functions μ and ν. The general approach is to find a set of constraints on these functions that are sufficient to guarantee that an S-patch form for H can be constructed subject to the G^1 continuity condition of Equation 6. Once the constraints have been determined, we construct μ and ν as the minimal degree polynomials that satisfy the constraints. It

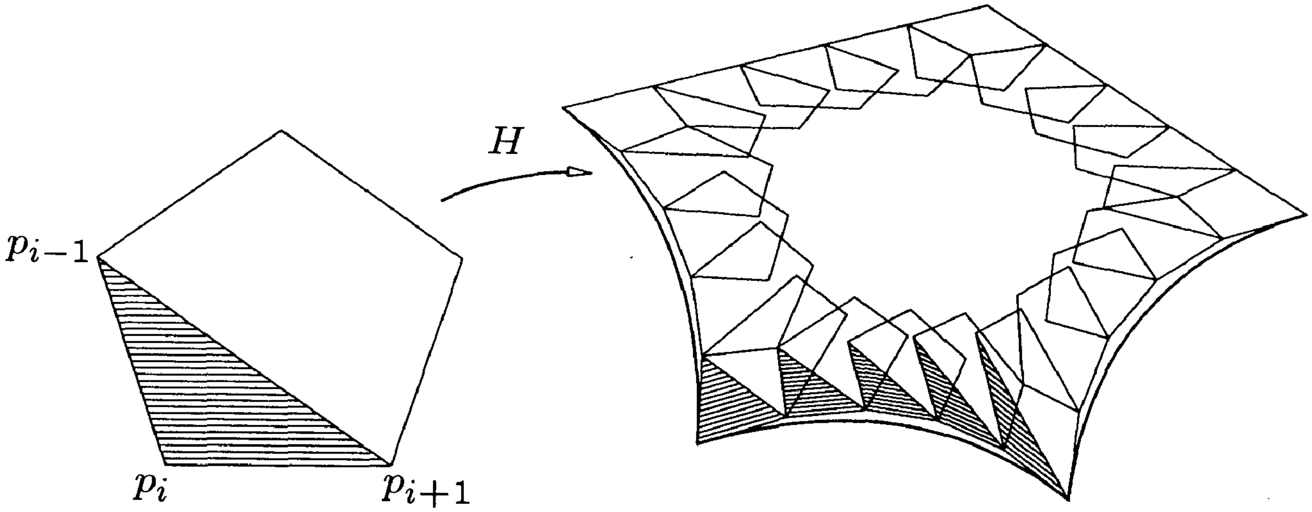

Figure 3: Boundary panels of the S-patch H.

turns out that six constraints are sufficient, and that the constraints are linear in the polynomial coefficients of μ and ν. By choosing μ and ν to be quadratic polynomials, six unknowns are introduced, and hence a unique solution to the system can be found. Our goal now is to determine these constraints.

REMARK : Equation 6 implies that the same functions μ, ν are used for every edge since μ and ν do not appear in the equation with an i subscript. This appears to be an unnecessary assumption. However, allowing μ and ν to differ from edge to edge introduces no new flexibility. Instead, what occurs is that a linear system of $6n$ equations in $6n$ unknowns is required, so again the solution is unique, and is such that the resulting functions are the same for each edge.

The first two constraints on μ and ν are determined by setting $t_i = 0$ in Equation 6, and using Equation 5 together with (A1) to give

$$\frac{\partial F_i}{\partial v_i}(0,1) = \mu(0)\frac{\partial F_i}{\partial u_i}(0,1) + \nu(0)\frac{\partial F_i}{\partial v_i}(0,1).$$

It is therefore sufficient for $\mu(0) = 0$ and $\nu(0) = 1$. These two constraints were found by restricting a condition imposed along E_i to the corner corresponding to p_i. The next two constraints follow by restricting to p_i a condition imposed along the edge E_{i-1}. To do this, we require an expression for $\mathbf{D}_{\vec{t}_i} H(E_{i-1}(t_{i-1}))$ in terms of F_{i-1}, F_i, μ and ν. The required expression can be found by noting that since P is a regular n-gon, $-\vec{t}_{i-1} = -2\cos\frac{2\pi}{n}\vec{t}_i + \vec{t}_{i+1}$. Linearity of directional differentiation therefore implies that

$$\mathbf{D}_{-\vec{t}_{i-1}} H = -2\cos\tfrac{2\pi}{n}\mathbf{D}_{\vec{t}_i} H + \mathbf{D}_{\vec{t}_{i+1}} H.$$

Substituting this expression into Equation 6, replacing i by $i-1$, and solving for $\mathbf{D}_{\vec{t}_i} H(E_{i-1}(t_{i-1}))$ yields

$$\begin{aligned}\mathbf{D}_{\vec{t}_i} H(E_{i-1}(t_{i-1})) &= (\mu(t_{i-1}) + 2\cos\tfrac{2\pi}{n})\frac{\partial F_{i-1}}{\partial u_{i-1}}(t_{i-1},1) \\ &\quad +\nu(t_{i-1})\frac{\partial F_{i-1}}{\partial v_{i-1}}(t_{i-1},1). \qquad (7)\end{aligned}$$

The restriction of this equation to the point p_i is achieved by setting t_{i-1} to one. Doing this, and using $\mathbf{D}_{\vec{t}_i} H(p_i) = \frac{\partial F_{i-1}}{\partial v_{i-1}}(0,1)$, allows us to deduce that it is sufficient for $\mu(1) = -2\cos\frac{2\pi}{n}$ and $\nu(1) = 1$.

The final two constraints on μ and ν follow from the fact that since H is sought in S-patch form, H must be twice differentiable everywhere on P. In particular, the order of differentiation should not matter, meaning that μ and ν must be constructed so that

$$\mathbf{D}_{\vec{t}_{i-1}}\mathbf{D}_{\vec{t}_i} H(p_i) = \mathbf{D}_{\vec{t}_i}\mathbf{D}_{\vec{t}_{i-1}} H(p_i). \qquad (8)$$

The left hand side of Equation 8 is expanded by differentiating Equation 6 with respect to t_{i-1}, and the right hand side is expanded by differentiating Equation 7 with respect to t_i. This process, together with (A3) is used to show that it is sufficient for $\mu'(0) = -\nu'(1)$ and $\mu'(1) = \nu'(0)$, where prime denotes the first derivative.

To summarize, μ and ν must satisfy the following constraints:

$$\begin{aligned} \mu(0) &= 0, & \nu(0) &= 1, \\ \mu(1) &= -2\cos\tfrac{2\pi}{n}, & \nu(1) &= 1, \\ \mu'(0) &= -\nu'(1), & \mu'(1) &= \nu'(0). \end{aligned}$$

This system is solved uniquely if μ and ν are assumed to be quadratic polynomials. In Bernstein form, the solution is

$$\begin{aligned}\mu(t) &= -\cos\tfrac{2\pi}{n}B_1^2(t) - 2\cos\tfrac{2\pi}{n}B_2^2(t), \\ \nu(t) &= B_0^2(t) + (1 - \cos\tfrac{2\pi}{n})B_1^1(t) + B_2^2(t),\end{aligned}$$

where $B_i^2(t) = \binom{2}{i}t^i(1-t)^{2-i}$. Closer inspection of μ shows that it is actually a linear polynomial. Additionally, notice that in the case $n = 4$, μ reduces to the zero polynomial and ν becomes identically one, indicating that H meets the surrounding four surfaces with strict C^1 continuity.

Having determined μ and ν as above, the position and differential of H around the perimeter of P are completely determined by Equations 5 and 6. H then can be represented in S-patch form on P as follows:

1. The restriction of H and its differential to each edge E_i is found in triangular Bernstein form on the triangle $p_{i-1}p_ip_{i+1}$. This requires only straightforward manipulation of Bernstein polynomials since each of the functions appearing in Equations 5 and 6 is known in Bernstein form. The result of this step is a collection of triangular panels along each edge, as shown in Figure 3. Specific formulas relevant to our constructions are given in Section 5.

2. For each triangular panel computed in step 1, and for each edge E_i, find the image of P under the affine map that carries $p_{i-1}p_ip_{i+1}$ to the vertices of the panel, as indicated in Figure 3. The polynomial representation algorithm for S-patches mentioned in Section 3 guarantees that the collection of interlocking n-gons thus formed represent H and its differential around the perimeter of P.

3. To complete the S-patch representation of H, the remainder of the S-patch control net, consisting of all *interior control points*, i.e. those that do not contribute to position or differential around the perimeter of P, must be determined. These points can therefore be set arbitrarily without influencing the G^1 join between H and the surrounding surfaces. Of course, an intelligent choice for the interior points must be made to avoid unwanted ripples in H away from the boundaries. A method for setting the interior points that works well in practice is given in Section 5.

5 Sabin nets

In preparation for the generalized B-spline schemes of Section 6, we recast the n-sided hole problem in terms of a self contained control point representation. We call this representation the *Sabin net*, due to its connection to the representations proposed by Sabin [17, 19] [‡]. Figures 4 and 5 illustrate quadratic and cubic Sabin nets respectively. Intuitively, Sabin nets are used to construct the boundary data needed by our solution to the n-sided hole problem. An S-patch is then constructed using the method of Section 4. The procedure to create an S-patch, given a Sabin net, is thus a three step process:

i) Construct boundary data from the Sabin net.

ii) Construct S-patch boundary panels as in Section 4.

iii) Construct the interior S-patch control points.

In Section 5.1, these steps are elaborated for the case of quadratic Sabin nets; Section 5.2 describes the case of cubic Sabin nets.

5.1 Quadratic case

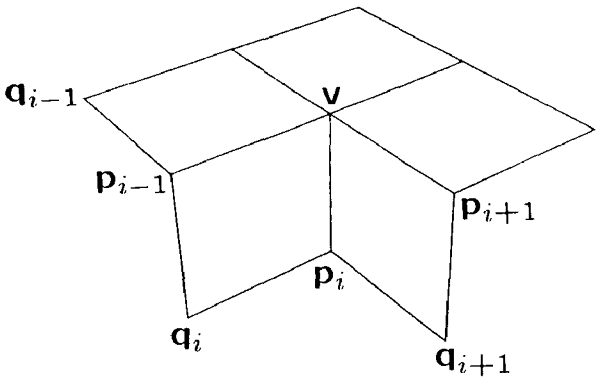

Figure 4: A quadratic Sabin net.

The first step is to convert the quadratic Sabin net into an instance of the n-sided hole problem. Assume the Sabin net labeling of Figure 4. The n boundary data functions are constructed as follows:

$$\begin{aligned}
F_i(u_i,1) &= \mathbf{q}_i B_0^2(u_i) + \mathbf{p}_i B_1^2(u_i) + \mathbf{q}_{i+1} B_2^2(u_i), \\
\frac{\partial F_i}{\partial v_i}(u_i,1) &= 2[(\mathbf{p}_{i-1} - \mathbf{q}_i) B_0^2(u_i) \\
&+ (\mathbf{v} - \mathbf{p}_i) B_1^2(u_i) + (\mathbf{p}_{i+1} - \mathbf{q}_{i+1}) B_2^2(u_i)].
\end{aligned}$$

It is not difficult to show that these functions satisfy conditions (A0) through (A3).

Next, triangular boundary panels of an S-patch H are computed by step 1 of Section 4. From Equation 6 we see that $\frac{\partial F_i}{\partial u_i}(u_i,1)$, $\frac{\partial F_i}{\partial v_i}(u_i,1)$, $\mu(t_i)$, and $\nu(t_i)$ are at most quadratic functions; this implies that $\mathbf{D}_{-\vec{t}_{i-1}} H(E(t_i))$ is at most quartic, and therefore $H(E(t_i))$ is at most quintic. This shows that the S-patch H must be at most depth 5. Let the $\mathbf{h}_{\vec{i}}$ denote control points of H. Working through the details of step 1, the following formulas are found for the boundary panels of H up to symmetries:

$$\begin{aligned}
\mathbf{h}_{5\vec{e}_i} &= \mathbf{q}_i \\
\mathbf{h}_{4\vec{e}_i+\vec{e}_{i+1}} &= \tfrac{3}{5}\mathbf{q}_i + \tfrac{2}{5}\mathbf{p}_i \\
\mathbf{h}_{3\vec{e}_i+2\vec{e}_{i+1}} &= \tfrac{3}{10}\mathbf{q}_i + \tfrac{6}{10}\mathbf{p}_i + \tfrac{1}{10}\mathbf{q}_{i+1} \\
\mathbf{h}_{3\vec{e}_i+\vec{e}_{i+1}+\vec{e}_{i-1}} &= \tfrac{2+2c}{5}\mathbf{q}_i + \tfrac{1-c}{5}\mathbf{p}_i + \tfrac{1-c}{5}\mathbf{p}_{i-1} + \tfrac{1}{5}\mathbf{v} \\
\mathbf{h}_{2\vec{e}_i+2\vec{e}_{i+1}+\vec{e}_{i-1}} &= \tfrac{7+8c}{30}\mathbf{q}_i + \tfrac{5+2c}{15}\mathbf{p}_i + \tfrac{1}{15}\mathbf{p}_{i+1} \\
&+ \tfrac{1-4c}{30}\mathbf{q}_{i+1} + \tfrac{1}{15}\mathbf{p}_{i-1} + \tfrac{4-4c}{15}\mathbf{v}
\end{aligned}$$

where $c = \cos\frac{2\pi}{n}$, and $i = 1 \ldots n$. The remaining boundary panel points of H may be found using step 2 of Section 4.

Constructing the boundary panels does not, in general, completely determine H. The remaining degrees of freedom are the positions of interior S-patch control points. The procedure for finding positions for the interior points of H is as follows:

1. Define a depth 2 S-patch A whose boundary control points match those of the Sabin net. Let $\mathbf{a}_{\vec{i}}$ denote the control points of A. This step is algorithmically established by

```
for i ← 1 to n
  $\mathbf{a}_{2\vec{e}_i} \leftarrow \mathbf{q}_i$
  $\mathbf{a}_{\vec{e}_i+\vec{e}_{i+1}} \leftarrow \mathbf{p}_i$
end for
```

2. The remaining control points of A are found as convex combinations of the control points of the Sabin net so that A approximates the desired shape of H. The following algorithm is used for this purpose

```
for i ← 1 to n
  for j ← 2 to $\lfloor \frac{n}{2} \rfloor$
    if odd(j) then
      $\mathbf{a}_{\vec{e}_i+\vec{e}_{(i+j)}} \leftarrow (1-\cos\frac{\pi j}{N})\mathbf{v} + \cos\frac{\pi j}{N}\mathbf{p}_{(i+\lfloor \frac{j}{2} \rfloor)}$
    else
      $\mathbf{a}_{\vec{e}_i+\vec{e}_{(i+j)}} \leftarrow (1-\cos\frac{\pi j}{N})\mathbf{v} + \cos\frac{\pi j}{N}\mathbf{q}_{(i+\lfloor \frac{j}{2} \rfloor)}$
    end if
  end for
end for.
```

3. The S-patch A is then depth elevated from depth 2 to 5 to match the depth of H.

[‡] The only difference is that we do not limit the number of sides.

4. The (unknown) interior control points of H are equated to the (known) interior control points of A.

The justification for this procedure is that the lower depth S-patch A has fewer unknown control points than H; it should therefore, be easier to find geometric constructions for these points, and the resulting surface should have fewer undulations than a higher depth patch.

It is not difficult to see that in the case $n = 4$, the Sabin net and the patch A are identical. The resulting surface is the usual tensor product biquadratic, although parameterized as a biquintic.

5.2 Cubic Boundary Case

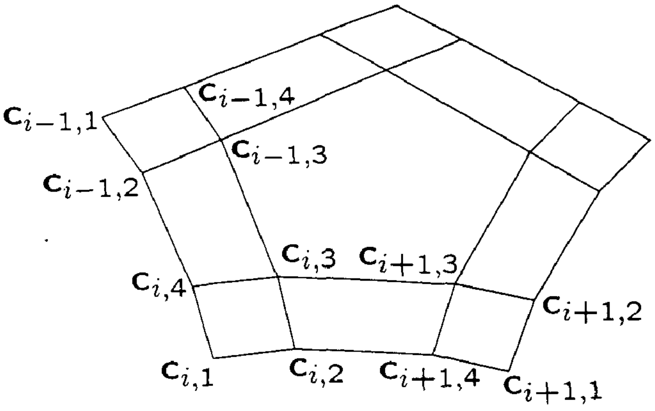

Figure 5: A cubic Sabin net.

Let the cubic Sabin net be labeled as in Figure 5. The n boundary functions of the n-sided hole problem are computed as follows:

$$\begin{aligned}
F_i(u_i, 1) &= \mathbf{c}_{i,1}B_0^3(u_i) + \mathbf{c}_{i,2}B_1^3(u_i) \\
&+ \mathbf{c}_{i+1,4}B_2^3(u_i) + \mathbf{c}_{i+1,1}B_3^3(u_i), \\
\frac{\partial F_i}{\partial v_i}(u_i, 1) &= 3[(\mathbf{c}_{i,4} - \mathbf{c}_{i,1})B_0^3(u_i) \\
&+ (\mathbf{c}_{i,3} - \mathbf{c}_{i,2})B_1^3(u_i) \\
&+ (\mathbf{c}_{i+1,3} - \mathbf{c}_{i+1,4})B_2^3(u_i) \\
&+ (\mathbf{c}_{i+1,2} - \mathbf{c}_{i+1,1})B_3^3(u_i)].
\end{aligned}$$

Once again, it is not difficult to verify that these functions also satisfy conditions (A0) through (A3).

The boundary panels of H are again computed by step 1 of Section 4. A degree argument similar to the one used in Section 5.1 shows that H must be a depth 6 S-patch. Completing step 1 leads to the following formulas:

$$\begin{aligned}
\mathbf{h}_{6\vec{e}_i} &= \mathbf{c}_{i,1} \\
\mathbf{h}_{5\vec{e}_1+\vec{e}_{i+1}} &= \tfrac{1}{2}\mathbf{c}_{i,1} + \tfrac{1}{2}\mathbf{c}_{i,2} \\
\mathbf{h}_{4\vec{e}_1+2\vec{e}_{i+1}} &= \tfrac{1}{5}\mathbf{c}_{i,1} + \tfrac{3}{5}\mathbf{c}_{i,2} + \tfrac{1}{5}\mathbf{c}_{i+1,4} \\
\mathbf{h}_{3\vec{e}_1+3\vec{e}_{i+1}} &= \tfrac{1}{20}\mathbf{c}_{i,1} + \tfrac{9}{20}\mathbf{c}_{i,2} + \tfrac{9}{20}\mathbf{c}_{i+1,4} + \tfrac{1}{20}\mathbf{c}_{i+1,1} \\
\mathbf{h}_{4\vec{e}_1+\vec{e}_{i+1}+\vec{e}_{i-1}} &= \tfrac{3}{10}\mathbf{c}_{i,3} + \tfrac{1-c}{5}\mathbf{c}_{i,4} + \tfrac{3+4c}{10}\mathbf{c}_{i,1} + \tfrac{1-c}{5}\mathbf{c}_{i,2} \\
\mathbf{h}_{3\vec{e}_1+2\vec{e}_{i+1}+\vec{e}_{i-1}} &= \tfrac{3-3c}{10}\mathbf{c}_{i,3} + \tfrac{1}{20}\mathbf{c}_{i,4} + \tfrac{3+4c}{20}\mathbf{c}_{i,1} \\
&+ \tfrac{3+3c}{10}\mathbf{c}_{i,2} + \tfrac{3}{20}\mathbf{c}_{i+1,3} + \tfrac{1-4c}{20}\mathbf{c}_{i+1,4}
\end{aligned}$$

where $c = \cos\frac{2\pi}{n}$ and $i = 1 \ldots n$. The remaining boundary panel points of H may be computed as in step 2 of Section 4.

The unknown interior points of H are computed by a procedure analogous to that of the quadratic case. Let A be a depth 3 S-patch. Again, the boundary control points of A are equated to the boundary control points of the Sabin net. This is stated algorithmically as

```
for $i \leftarrow 1$ to $n$
    $\mathbf{a}_{2\vec{e}_i+\vec{e}_{i-1}} \leftarrow \mathbf{c}_{i,4}$
    $\mathbf{a}_{3\vec{e}_i} \leftarrow \mathbf{c}_{i,1}$
    $\mathbf{a}_{2\vec{e}_i+\vec{e}_{i+1}} \leftarrow \mathbf{c}_{i,2}$
end for
```

Next, we compute the remaining control points of A. Let the Sabin net control points $\mathbf{c}_{1,3}, \mathbf{c}_{2,3}, \ldots, \mathbf{c}_{n,3}$ be treated as the control points of a depth 1 S-patch $\tilde{A}$. Let $w_{\vec{i}}$ be defined by

$$w_{\vec{i}} = \frac{i_1 p_1 + i_2 p_2 + \cdots + i_n p_n}{n},$$

where $|\vec{i}| = 3$. The domain of $\tilde{A}$ is taken to be the polygon $w_{\vec{e}_n+\vec{e}_1+\vec{e}_2}, w_{\vec{e}_1+\vec{e}_2+\vec{e}_3}, \ldots, w_{\vec{e}_{n-1}+\vec{e}_n+\vec{e}_1}$. The remaining control points of A are found as follows

```
for all $w_{\vec{i}} \notin$ boundary($P$)
    $\mathbf{a}_{\vec{i}} \leftarrow \tilde{A}(w_{\vec{i}})$
end for
```

The surface represented by A is now depth elevated from 3 to 6, and the (unknown) interior points of H are equated to the (known) interior points of A.

In the case $n = 4$, A and the Sabin net are identical, implying that the resulting patch H is a tensor product bicubic, parameterized as a bisextic.

6 Generalized B-spline Schemes

We now present two schemes for modeling surfaces with arbitrary topology: one is an generalization of biquadratic B-splines , the other a generalization of bicubic B-splines. Like traditional B-spline methods, these schemes take as input a control mesh that roughly approximate the desired shape, producing as output a smooth spline surfaces.

These schemes are inspired by the close relationship between the control points of B-spline and Bézier representations. The essence of this relationship is that the Bézier control points may be found by local averaging of the B-spline control points. Such a construction was first presented by Sablonniére[20] for curves, and later by Boehm [1, 2] for B-spline surfaces. Here, we present analogous constructions for computing Sabin net control points by a local averaging of the input control mesh.

Adjacent Sabin nets are constructed to share the same boundary position function, $F_i(u_i, 1)$, and cross boundary tangent vector function $\frac{\partial F_i}{\partial v_i}(u_i, 1)$ (up to a constant scale factor), corresponding to the common boundary. The resulting S-patches, constructed by the method of Section 5, are thus guaranteed to meet with G^1 continuity.

The conditions that adjacent Sabin nets must satisfy in order to share the same boundary data functions are easily described. Equivalent boundary position for adjacent Sabin nets is achieved by sharing boundary points. To share equivalent cross boundary tangent vectors, the three rows of control points parallel to the boundary must be co-linear and form identical affine segments (see Figures 6 and 7). These conditions are equivalent to the usual C^1 conditions for Bézier tensor product patches.

6.1 Generalized Biquadratic Scheme

A control mesh for the generalized biquadratic scheme is restricted to 4-sided faces, but a vertex can be shared by any number of faces. There is a one to one correspondence between vertices of the control mesh and individual patches of the spline surface. That is, for each vertex that is shared by n faces, an n-sided, quadratic Sabin net is constructed.

The Sabin net corresponding to a vertex is found by constructing the midpoints of all edges incident on the vertex, and the centroids of all incident faces. These points, together with the vertex form the Sabin net (see Figure 6). It is easy to verify that neighboring Sabin nets satisfy the continuity constraints. This construction is equivalent to that purposed by Sabin [19], but is not limited to patches with at most 5 sides.

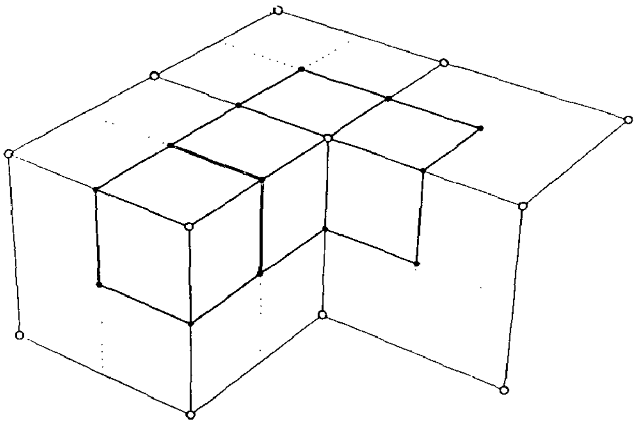

Figure 6: 3-sided and 5-sided quadratic Sabin nets are constructed to satisfy the continuity constraints.

The Sabin nets are converted to S-patches by the method given in Section 5. Figures 8 and 9 show some examples of spline surfaces constructed using this scheme.

6.2 Generalized Bicubic Scheme

A control mesh for the generalized bicubic scheme may have faces with any number of sides, but each (non-boundary) vertex must belong to exactly four faces. In this scheme, there is a one to one correspondence between faces of the control mesh and patches of the spline surface. That is, for each n-sided face of the control mesh, an n-sided cubic Sabin net is produced.

Generating the Sabin net is slightly more complicated than for the generalized biquadratic scheme given in Section 6.1. In this case n *key points* are found for each n-sided face of the control mesh. The key points are connected across edges to form quadrilaterals corresponding to the vertices of the control mesh. The points of the Sabin net are taken to be edge midpoints and the centroids of the quadrilaterals as well as the key points (see Figure 7).

The positioning of the key points can greatly affect the final shape of the surface. One , ossibility might be to leave them as user specified *shape handles.* Instead, we propose reasonable default placements for the key points. Let the vertices and centroid of a face be labeled as in Figure 7. The key point $\mathbf{k}_i$ is computed by

$$\mathbf{k}_i = \tfrac{1}{9}\mathbf{b}_{i-1} + \tfrac{3}{9}\mathbf{b}_i + \tfrac{1}{9}\mathbf{b}_{i+1} + \tfrac{4}{9}\mathbf{c}$$

where $\mathbf{c}$ is the centriod of the face $\mathbf{b}_1, \ldots, \mathbf{b}_n$. This choice is justified empirically, and it generalizes the construction of bicubic B-splines in the case $n = 4$. Other possiblities certainly exist.

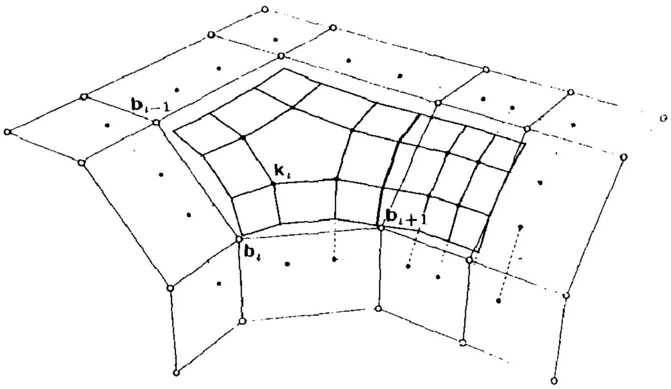

Figure 7: 4-sided and 5-sided cubic Sabin nets are constructed subject to the continuity constrains. Filled dots represent key points.

Once the Sabin nets have been constructed, they are converted to S-patches by the method given in Section 5. Figures 10 and 11 show some examples of spline surfaces constructed using this scheme.

7 Conclusions

We have presented two methods for constructing surfaces from control meshes that provide sufficient generality for the modeling of arbitrary topological surfaces. One of the methods is a generalization of biquadratic B-splines, the other a generalization of bicubic B-splines. We have based these generalizations on S-patches, a class of n-sided parametric surface patches that contains Bézier triangular and tensor product patches as proper subsets. Our methods are also based on well known geometric constructions for converting from a B-spline to Bézier representation: we convert a more general control mesh into a collection of S-patches that meet with tangent plane continuity.

The control meshes of this paper are not entirely general in that we require that exactly four surface patches meet at each interior vertex of the spline surface. This constraint was imposed for reasons of simplicity; the so called "twist compatibility" problem has an easily described solution in this case. More general patch connections are certainly possible, and are currently being developed by the authors.

Other topics of future research include: finding better ways of determining the interior control points of for the S-patch construction of Section 5, experimenting with alternatives to the key point placement algorithm of Section 6.2, and extending these results to produce curvature continuous (G^2) spline surfaces.

We have not considered the case of Non-Uniform Rational B-splines or NURBs. The main reasons for this omission are the desire to avoid unnecessary complications, and because "knot lines" do not seem to have well defined counterparts when dealing with arbitrary topologies. However, our constructions are easily extended to include non-uniformity by replacing the word "midpoint" with "knot vector ratio" in the constructions of Section 6. It is also quite straightforward to incorporate rational polynomials through the use of homogeneous coordinates.

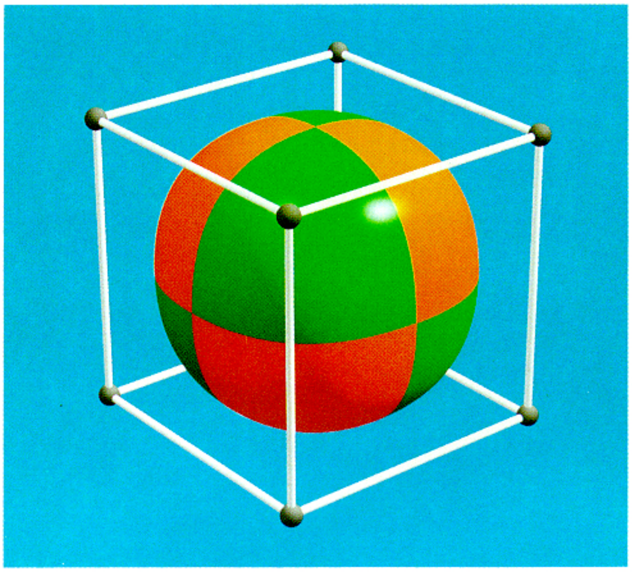

Figure 8: A closed surface produced by the generalized biquadratic scheme.

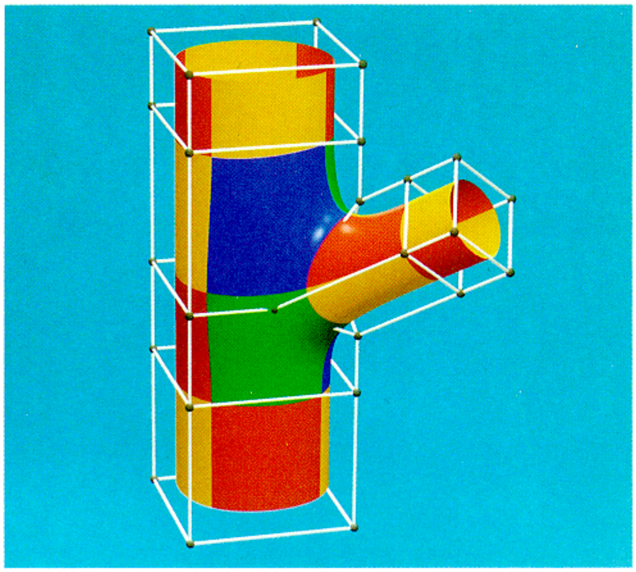

Figure 10: A branching surface produced by the generalized bicubic scheme.

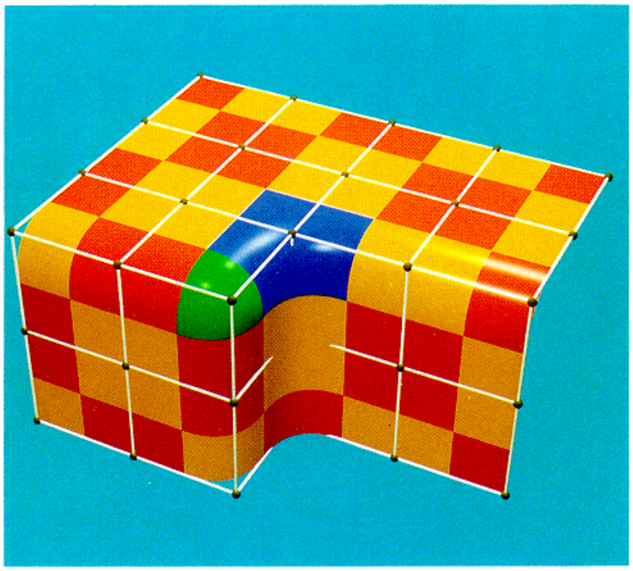

Figure 9: A surface containing fillets and blends produced by the generalized biquadratic scheme.

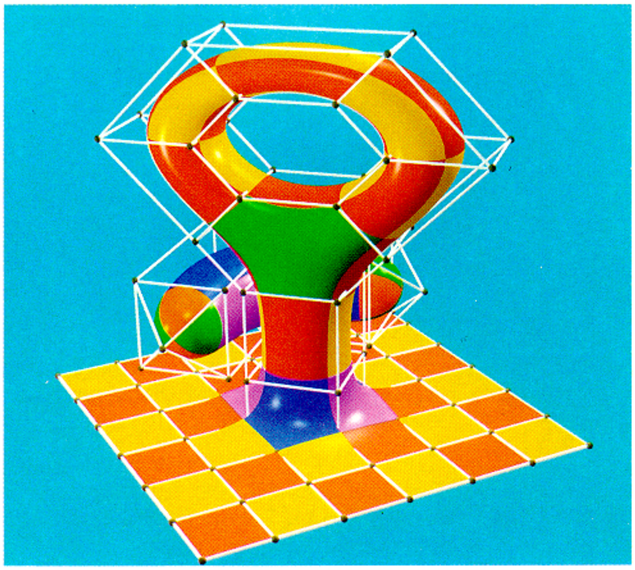

Figure 11: A surface with a handle and a closed surface produced by the generalized bicubic scheme.

References

[1] Boehm, Wolfgang. Cubic B-spline curves and surfaces in computer aided geometric design. *Computing*, 19:29–34, 1977.

[2] Boehm, Wolfgang. Generating the Bézier points of B-splines. *Computer Aided Design*, 13(6):365–366, 1981.

[3] Boehm, Wolfgang. Visual continuity. *Computer Aided Design*, 20(6):307–311, 1988.

[4] Catmull, Edwin and James Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10(6):350–355, 1978.

[5] Charrot, Peter and John Gregory. A pentagonal surface patch for computer aided geometric design. *Computer Aided Geometric Design*, 1(1):87–94, 1984.

[6] Chiyokura, Hiroaki and Fumihiko Kimura. Design of solids with free-form surfaces. *Computer Graphics*, 17(3):289–298, 1983.

[7] de Boor, Carl. B-form basics. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 131–148. SIAM, 1987.

[8] DeRose, Tony. *Geometric Continuity: A Parametrization Independent Measure of Continuity for Computer Aided Geometric Design*. PhD thesis, Berkeley, 1985. also available as Technical report UCB/CSD 86/255.

[9] Doo, Daniel and Malcolm Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer Aided Design*, 10(6):356–360, 1978.

[10] Gregory, John and Jörg Hahn. A C^2 polygonal surface patch. *Computer Aided Geometric Design*, 6(1):69–75, 1989.

[11] Gregory, John. N-sided surface patches. In J. Gregory, editor, *The Mathematics of Surfaces*, pages 217–232. Clarendon Press, 1986.

[12] Guibas, Leo and Jorge Stolfi. Primitives for the manipulation of general subdivisions and the computation of voronoi diagrams. *ACM Transactions on Graphics*, 4(2):74–123, 1985.

[13] Hahn, Jörg Filling polygonal holes with rectangular patches. In W. Strasser and H.P. Seidel, editors, *Geometric Modeling: Algorithms and New Trends*, pages 81–91. Spring-Verlag, 1989.

[14] Herron, Gary. *Triangular and Multisided Patch Schemes*. PhD thesis, U. of Utah, 1979.

[15] Herron, Gary. Smooth closed surfaces with discrete triangular interpolants. *Computer Aided Design*, 2(4):297–306, 1985.

[16] Herron, Gary. Techniques for visual continuity. In G. Farin, editor, *Geometric Modeling*, pages 163–174. SIAM, 1987.

[17] Hosaka, Mamoru and Fumihiko Kimura. Non-four-sided patch expressions with control points. *Computer Aided Geometric Design*, 1(1):75-86, 1984.

[18] Loop, Charles and Tony DeRose. A multisided generalization of Bézier surfaces. *ACM Transactions on Graphics*, 8(3):204–234, 1989.

[19] Sabin, Malcolm. Non-rectangular surface patches suitable for inclusion in a B-spline surface. In P. ten Hagen, editor, *Proceedings of Eurographics '83*, pages 57–69. North-Holland, 1983.

[20] Sablonniere, Paul. Spline and Bézier polygons associated with a polynomial spline curve. *Computer Aided Design*, 10(4):257–261, 1978.

[21] van Wijk, Jarke. Bicubic patches for approximating non-rectangular control-point meshes. *Computer Aided Geometric Design*, 3(1):1–13, 1986.

[22] Varady, Tamas. Survey and new results in n-sided patch generation. In R. Martin, editor, *The Mathematics of Surfaces II*, pages 203–236. Oxford University Press, 1987.