Smooth Spline Surfaces over Irregular Meshes

Charles Loop
Apple Computer, Inc.

Abstract

An algorithm for creating smooth spline surfaces over irregular meshes is presented. The algorithm is a generalization of quadratic B-splines; that is, if a mesh is (locally) regular, the resulting surface is equivalent to a B-spline. Otherwise, the resulting surface has a degree 3 or 4 parametric polynomial representation. A construction is given for representing the surface as a collection of tangent plane continuous triangular Bézier patches. The algorithm is simple, efficient, and generates aesthetically pleasing shapes.

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, Triangular patches, Geometric continuity, Irregular meshes, Aribitrary topology.

1 Introduction

The B-spline paradigm for modeling smooth surfaces is limited by the requirement that the control point mesh must be organized as a regular rectangular structure. Ignoring this requirement by collapsing control mesh edges leads to surfaces with ambiguous surface normals and degenerated parameterizations. A more general method is to construct a surface from a mesh of points without degeneracy. By constructing this surface using piecewise polynomials, familiar algebraic tools can be brought to bear for analysis. This is the approach taken in this paper.

A new type of spline surface is presented for modeling surfaces of arbitrary topological type by smoothly approximating an irregular control mesh. The advantage of this technique over existing schemes is simplicity, efficiency, and piecewise polynomial form. The spline surface is simply constructed by computing a triangular Bézier representation of a network of surface patches. Being of fairly low polynomial degree (at most 4), these patches are efficient to compute and evaluate. The spline surface is smooth, since the patches fit together with tangent plane continuity. Another advanage of this scheme is a close relationship to quadratic B-spline surfaces. In regular regions of the mesh, the surface is equivalent to a B-spline represented by bi-quadratic Bézier patches. This property can represent a considerable savings in time and space, since in practice control meshes often have few irregularities.

The spline algorithm takes an irregular control mesh as input. A new refined mesh is created with more faces, vertices, and edges than the original. The new mesh has a simpler structure since every vertex has exactly four edges incident upon it. Next, an intermediate form called a "quad-net" is constructed corresponding to each vertex of the refined mesh. The quad-nets characterize local 4-sided regions of the surface in a uniform way. Finally, a group of four quartic triangular patches are constructed for each quad-net as output. The union of these patches constitutes a smooth spline surface.

This paper is organized as follows: previous work is surveyed in § 2. Relevant background material, including Bézier forms and B-splines are covered in § 3. The spline algorithm is presented as a sequence of pipeline stages in § 4. A detailed development of the smoothness constraints used to construct the surface is presented in § 5. Concluding remarks are found in § 6. Special techniques for dealing with meshes with boundaries (i.e., meshes that are not closed) are given in Appendix A.

2 Previous Work

The earliest attempts to overcome the topological limitations of B-spline surfaces were based on the refinement principle[1, 4]. The idea is to refine, or subdivide, an irregular mesh by creating a new mesh, with more faces and vertices, that approximates the old. By repeating this process, a smooth surface is formed in the limit. Subdivision algorithms are conceptually quite simple and generally generate nice shapes. However, subdivision surfaces do not admit an analytic form, complicating their use in many practical applications. Despite this, algorithms based on subdivision surfaces continue to appear[7].

Gregory patches have been used to interpolate the vertices of an irregular mesh[2]. These patches have singularities at corners and are not polynomial. Other non-polynomial surface patches used to define B-spline-like surfaces over irregular meshes include the 3 and 5-sided patches defined in[17], and n-sided S-patches[9, 12]. S-patch based schemes can be inefficient (in both time and space) for $n > 5$ or 6. A generalization of quartic triangular B-splines to strictly triangular meshes using degree six polynomial patches appears in [11]. Other schemes use tensor

Author's address: Apple Computer, Inc., 1 Infinite Loop, MS:301-3J, Cupertino, CA 95014; loop@apple.com

product polynomials, but require the connectivity of the control mesh to be restricted[6, 18].

More recent approaches to the problem assume that irregular vertices (a vertex with other than 4 edges incident upon it) are isolated. That is, every irregular vertex is surrounded by one or more layers of quadrilaterals and regular vertices. G-splines[8, 16] take this approach. Severals schemes by Peters[13, 14, 15] isolate irregularities by applying one or more refinement steps to an irregular mesh. The approach taken in this paper is similar. The distinguishing features are that only one refinement step is required, and the mesh does not have to be preprocessed to have excusively 3 or 4-sided faces. The trade-off for this simplification is fewer patches of higher degree. The patches computed here are at most polynomial degree 4, as opposed to degree 3 in [10, 13, 14].

3 Background

This section gives a brief review of Bézier curves and surfaces, and B-spline surfaces. Consult [5] for additional details.

3.1 Bézier forms

A degree d *Bézier curve* is defined

$$B(t) = \sum_{i=0}^{d} \mathbf{b}_i B_i^d(t),$$

where $t \in [0,1]$, the points $\mathbf{b}_i$ form the Bézier *control polygon*, and

$$B_i^d(t) = \binom{d}{i}(1-t)^{d-i}t^i,$$

are degree d Berstein polynomials. As a convenient notation, a Bézier curve will be identified by its control polygon represented by the vector $[\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_d]$.

A degree r by s *tensor product Bézier patch* is defined

$$B(u,v) = \sum_{i=0}^{r}\sum_{j=0}^{s} \mathbf{b}_{ij} B_i^r(u) B_j^s(v),$$

where $u, v \in [0,1]$, and the $\mathbf{b}_{ij}$ are a rectangular array of points forming the Bézier *control net*.

A degree d *Bézier triangle* is defined

$$B(u,v) = \sum_{i+j+k=d} \mathbf{b}_{ijk} B_{ijk}^d(u,v),$$

where $u, v, (1-u-v) \in [0,1]$, ij, and k are non-negative integers that sum to d, the $\mathbf{b}_{ijk}$ form a triangular Bézier control net, and

$$B_{ijk}^d(u,v) = \binom{d}{ijk}(1-u-v)^i u^j v^k,$$

are the degree d bi-variate Bernstein polynomials where $\binom{d}{ijk}$ is the trinomial coefficient $\frac{d!}{i!j!k!}$.

Bézier surfaces are a convenient representation for individual polynomial patches. Algorithms for rendering, raytracing, and surface intersection often utilize the Bézier form. When constructing smooth composite surfaces consisting of several patches, satisfying the necessary smoothness constraints among Bézier surfaces can be quite complex. In this setting, it is preferable to use B-spline surfaces.

3.2 B-splines

A *tensor product B-spline surface* is defined

$$S(u,v) = \sum_i \sum_j \mathbf{d}_{ij} N_i^r(u) N_j^s(v),$$

where the $\mathbf{d}_{ij}$ form a rectangular control mesh and the N_k^d are order d (degree $d-1$) B-spline *basis functions*. Each basis function is defined over a partition of the real axis called a *knot vector* (see [5] for details).

Two particular properties of B-splines are of interest here. First, by introducing a new knot between each pair of existing knots, the control mesh is refined, or subdivided, without changing the shape of the surface. Second, B-splines are piecewise polynomial, therefore it is possible to represent a B-spline surface as a collection of individual polynomial patches. The spline surface presented in this paper is closely related to quadratic B-spline surfaces with uniform knots.

A quadratic B-spline can be represented as a composite of bi-quadratic tensor product Bézier patches. A single such patch is constructed corresponding to each vertex $\mathbf{d}_{ij}$ of the control mesh as illustrated in Figure 1. The corner points $\mathbf{b}_{00}$, $\mathbf{b}_{20}$, $\mathbf{b}_{02}$, and $\mathbf{b}_{22}$ are found as the centroids of the four faces surrounding $\mathbf{d}_{ij}$. The points $\mathbf{b}_{10}$, $\mathbf{b}_{01}$, $\mathbf{b}_{21}$, and $\mathbf{b}_{12}$ are found as midpoints of the four edges incident on $\mathbf{d}_{ij}$, and the point $\mathbf{b}_{11}$ is equivalent to $\mathbf{d}_{ij}$.

The refinement algorithm for quadratic B-splines involves computing a new vertex corresponding to each {vertex, face} pair of the original mesh. The new vertices are found as weighted averages of the points belonging to each face of the original mesh. For the quadratic B-spline case, these weights (going around a face) are $\{\frac{9}{16}, \frac{3}{16}, \frac{1}{16}, \frac{3}{16}\}$. The newly created vertices are then connected to form the faces of the refined control mesh.

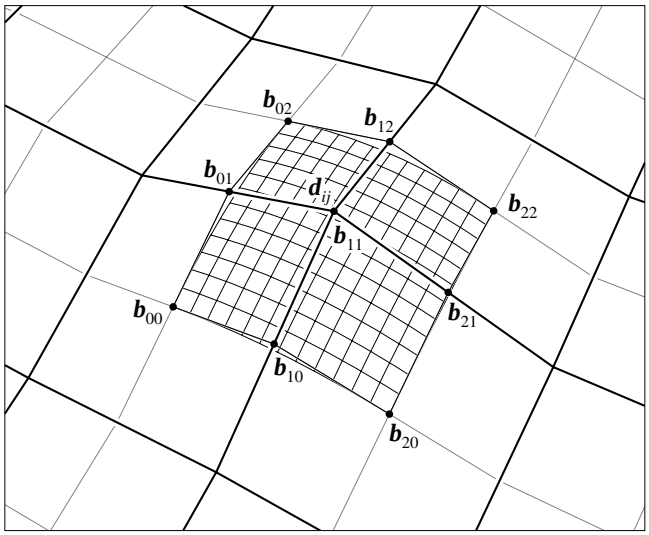


Figure 1: The bi-quadratic Bézier patch corresponding to the B-spline control mesh point $\mathbf{d}_{ij}$.

A B-spline surface is smooth because adjacent patches share positions and first derivatives at all points along common boundaries. This notion of matching derivatives along patch boundaries is sufficient because the domain of each patch lies in a single uv plane. Therefore, a B-spline surface is a deformation of this domain plane. For this reason, B-spline surfaces can only model shapes that are topologically planar[†].

[†]B-spline surfaces may also be defined over cylinders and tori, as these domains can tile the plane.

Under a less restrictive definition, a surface is considered *smooth* if at all points it has a continuous, well-defined tangent plane. This notion is known as first order *geometric continuity*[3] and denoted G^1. In the next section, a spline surface is created by constructing a collection of patches over independent domains such that the union of this collection is G^1

4 Constructing the Spline

Constructing the spline surface begins with a user-defined *control mesh* denoted $\mathcal{M}^0$. A control mesh is a collection of vertices, edges, and (not necessarily planar) faces that can intuitively be thought of as a polygonal surface that may, or may not, be closed.[‡] The term *valance* is used to denote the number of edges incident on a vertex.

The spline surface is constructed in the following stages:

Input: *irregular control mesh*

1. refine mesh
2. construct quad-nets
3. construct patches

Output: *collection of triangular patches*

The mesh $\mathcal{M}^0$ is passed to a refinement procedure that creates a new mesh $\mathcal{M}^1$. The purpose of the refinement procedure is to isolate irregularities. After the refinement step, the mesh $\mathcal{M}^1$ is used to construct a set of *quad-nets*. The quad-nets characterize the surface locally, and provide a uniform structure for the third and final step. From each quad-net, a collection of four quartic triangular Bézier patches is constructed and output. The details of each step are described in the next three sections, followed by some examples.

4.1 Mesh Refinment

The first step takes a user-defined control mesh $\mathcal{M}^0$ and creates a new refined mesh $\mathcal{M}^1$. The vertices of $\mathcal{M}^1$ are constructed to corresponded to each {vertex, face} pair of $\mathcal{M}^0$. Let F be a face of $\mathcal{M}^0$ consisting of vertices $\{P_0, P_1, \ldots, P_{n-1}\}$ with centroid O (the average of the P_i's). The point P_i' of $\mathcal{M}^1$ corresponding to $\{P_i, F\}$ is found by

$$P_i' = \tfrac{1}{4}O + \tfrac{1}{8}P_{i-1} + \tfrac{1}{2}P_i + \tfrac{1}{8}P_{i+1},$$

where all subscripts are taken modulo n.

The faces of $\mathcal{M}^1$ are constructed corresponding to a vertex, face, or edge of $\mathcal{M}^0$. Each k-valant vertex of $\mathcal{M}^0$ will generate a k-sided face belonging to $\mathcal{M}^1$. Similarly, each n-sided face of $\mathcal{M}^0$ will generate an n-sided face belonging to $\mathcal{M}^1$. Finally, each edge of $\mathcal{M}^0$ will generate a 4-sided face belonging to $\mathcal{M}^1$. This construction is illustrated in Figure 2. Note that all the vertices of $\mathcal{M}^1$ are 4-valant, and every non-4-sided face is surrounded by 4-sided faces. Special consideration for vertices and edges that belong to the boundary of $\mathcal{M}^0$ can be found in Appendix A.

Remark : The refinement rule given here is equivalent to quadratic B-spline refinement for regular meshes. A more general construction of the refined mesh points due to Peters[13] associates a pair of scalar values u and v with each point P_i' such that

$$P_i' = (1-u)(1-v)O + \tfrac{(1-u)v}{2}P_{i-1} + \tfrac{u+v}{2}P_i + \tfrac{u(1-v)}{2}P_{i+1}.$$

The parameters u and v are similar to knots of a B-spline in that they may be used to locally adjust the shape of the surface.

[‡]More technically, a control mesh is a tessellated, oriented 2-manifold (possibly with boundary).

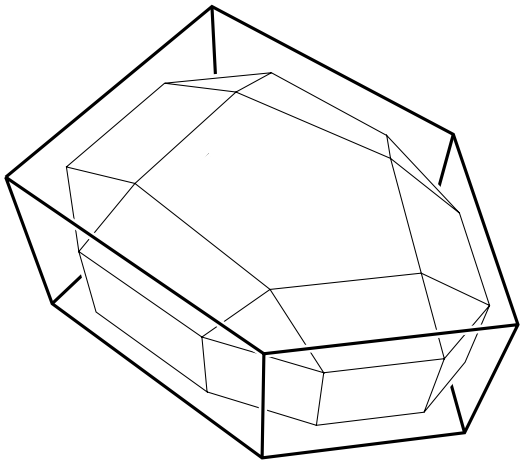


Figure 2: Mesh refinement: The vertices of the refined mesh $\mathcal{M}^1$ (thin lines) correspond to {vertex,face} pairs of the original mesh $\mathcal{M}^0$ (bold lines).

4.2 Quad-Nets

In the second step, 16 points and a pair of integers collectively referred to as a *quad-net* are constructed corresponding to each vertex of $\mathcal{M}^1$. Though quad-nets are in many ways like the control nets of Bézier patches, their purpose here is only as an intermediate stage between the refined mesh and the final triangular Bézier surface patches. A quad-net and its labeling scheme are illustrated in Figure 3.

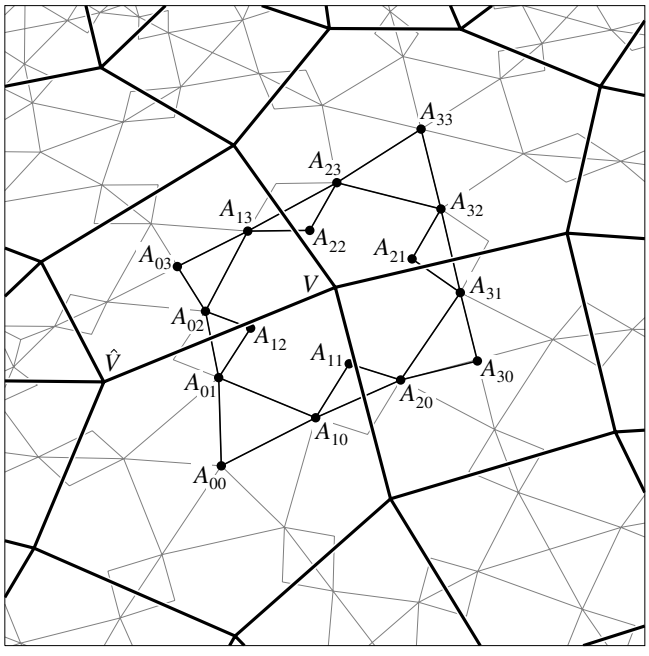


Figure 3: The quad-net corresponding to the vertex V of $\mathcal{M}^1$.

A quad-net locally characterizes a piece of the spline surface bounded by the four cubic Bézier curves $[A_{00}, A_{10}, A_{20}, A_{30}]$, $[A_{30}, A_{31}, A_{32}, A_{33}]$, $[A_{33}, A_{23}, A_{13}, A_{03}]$, and $[A_{30}, A_{02}, A_{01}, A_{00}]$. The corners A_{00}, A_{30}, A_{03} and A_{33} lie at the centroids of the four faces surrounding a vertex V. The interior points A_{11}, A_{12}, A_{21},

and A_{22} help specify the tangent plane along each of the four boundary curves.

In order to ensure that the spline surface is G^1, some constraints must be satisfied between the points of a pair of adjacent quad-nets. These constraints are as follows:

$$(1-c)A_{00} + cA_{01} = \tfrac{1}{2}A_{10} + \tfrac{1}{2}\hat{A}_{10}, \quad (1)$$

$$\tfrac{1}{2}A_{01} + \tfrac{1}{2}A_{02} = \tfrac{1}{2}A_{12} + \tfrac{1}{2}\hat{A}_{12}, \quad (2)$$

$$A_{03} = \tfrac{1}{2}A_{13} + \tfrac{1}{2}\hat{A}_{13}, \quad (3)$$

where $\hat{A}_{10}$, $\hat{A}_{12}$ and $\hat{A}_{13}$ are points of an adjacent quad-net, and c is a scalar to be determined. Similar constraints apply for the other three boundary curves in a symmetric manner. Justification for Constraints (1), (2), and (3) is provided in § 5.

Constraint (1) must hold between all pairs of adjacent quad-nets that share the point A_{00}. This implies that all quad-net points surrounding A_{00} must be co-planar. The following theorem is the key to constructing quad-net points that satisfy this requirement:

Theorem 4.1 *Let $P_0, \ldots, P_{n-1} \in \Re^3$ be a set of points in general position. The set of points $Q_0, \ldots, Q_{n-1}$ found by*

$$Q_i = \tfrac{1}{n}\sum_{j=0}^{n-1} P_j(1 + \beta(\cos\tfrac{2\pi(j-i)}{n} + \tan\tfrac{\pi}{n}\sin\tfrac{2\pi(j-i)}{n})), \quad (4)$$

satisfy

$$(1-\cos\tfrac{2\pi}{n})O + \cos\tfrac{2\pi}{n}Q_i = \tfrac{1}{2}Q_{i-1} + \tfrac{1}{2}Q_{i+1}, \quad (5)$$

where

$$O = \tfrac{1}{n}\sum_{j=0}^{n-1} P_j,$$

and are therefore co-planar.

Proof : See Appendix B. □

The factor β in Equation (4) is a free parameter that may be set arbitrarily. Theorem 4.1 applies to the construction at hand by setting

$$\beta = \tfrac{3}{2}(1 + \cos\tfrac{2\pi}{n}),$$

and interpreting the points $P_0, \ldots, P_{n-1}$ as the vertices of a face belonging to mesh $\mathcal{M}^1$, the point O as A_{00}, and the points Q_0, …, Q_{n-1} as the quad-net points surrounding A_{00}. Under this interpretation it is immediately clear from (5) that Constraint (1) is satisfied with $c = \cos\frac{2\pi}{n}$. Constructing the points A_{30}, A_{03}, and A_{33} and the surrounding quad-net points is similar. The observation that every n-sided face of $\mathcal{M}^1$ ($n \neq 4$) is surrounded by 4-sided faces, indicates that faces containing A_{30} and A_{03} are always 4-sided. Constraint (3) is satisfied since $\cos\frac{2\pi}{n} = 0$ when $n = 4$.

Applying Theorem 4.1 to each of the four faces surrounding a vertex of $\mathcal{M}^1$ will produce all of the quad-net points except for the four interior points A_{11}, A_{12}, A_{21}, and A_{22}. The construction for the point A_{12} is as follows: let V be the vertex about which the quad-net is constructed, and let $\hat{V}$ be an edge sharing neighbor of V (see Figure 3). Compute

$$A_{12} = \tfrac{1}{2}A_{01} + \tfrac{1}{2}A_{02} + \tfrac{1}{6}(V - \hat{V}), \quad (6)$$

and by symmetry

$$\hat{A}_{12} = \tfrac{1}{2}A_{01} + \tfrac{1}{2}A_{02} + \tfrac{1}{6}(\hat{V} - V). \quad (7)$$

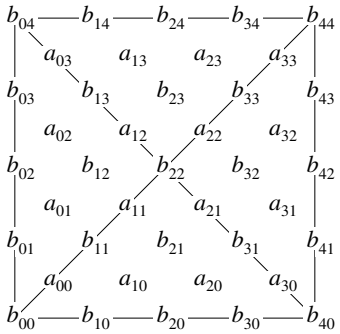
Averaging Equations (6) and (7) shows that Constraint (2) is satisfied. The construction of the other three interior quad-net points is symmetric.

The sixteen quad-nets points do not by themselves give enough information to construct surface patches that meet neighboring patches smoothly. The pair of integers n_0 and n_1 that correspond to the number of sides belonging to the faces that contain points A_{00} and A_{33} respectively are also needed. These two integers characterize the relationship between a quad-net and its neighbors when $\cos\frac{2\pi}{n_0}$ or $\cos\frac{2\pi}{n_1}$ are substituted for c in Constraint (1). The quad-nets are now passed to the next step where patches are constructed.

4.3 Constructing Patches

In the third and final step, parametric surface patches are constructed that interpolate the information encoded by the quad-nets constructed in step 2. A single bi-cubic patch is not sufficient to interpolate this data in general, since the mixed partial or *twist* terms at the corners of a quad-net may not be consistent (i.e., $\frac{\partial^2}{\partial u\,\partial v} \neq \frac{\partial^2}{\partial v\,\partial u}$, where u and v correspond to boundary curve parameters).

This difficulty can be eliminated by using four triangular patches that form an 'X' with respect to the four quad-net boundary curves. Cubic triangular patches suffice to interpolate the quad-net boundary curves, but do not have enough degrees of freedom to satisfy smoothness constraints across quad-nets boundaries. By using quartic patches, additional degrees of freedom are introduced that can be used to ensure smooth joins between adjacent triangular patches. The labeling scheme used for the Bézier control nets of the four quartic patches is as follows:



Formulas for the Bézier control points of one of the triangular patches are now given. Similar formulas for the other three patches can be found by symmetry. Interpolating the cubic boundary curves of a quad-net is achieved by degree raising, resulting in

$$\begin{aligned} b_{00} &= A_{00}, \\ b_{01} &= \tfrac{1}{4}A_{00} + \tfrac{3}{4}A_{01}, \\ b_{02} &= \tfrac{1}{2}A_{01} + \tfrac{1}{2}A_{02}, \\ b_{03} &= \tfrac{3}{4}A_{02} + \tfrac{1}{4}A_{03}, \\ b_{04} &= A_{03}. \end{aligned}$$

Tangent plane continuity is maintained across quad-net boundaries by setting

$$\begin{aligned} a_{00} &= \tfrac{1}{2}b_{10} + \tfrac{1}{2}b_{01}, \\ a_{01} &= \tfrac{c}{8}A_{00} + \tfrac{3-3c}{8}A_{01} + \tfrac{c}{4}A_{02} + \tfrac{1}{8}A_{10} + \tfrac{1}{2}A_{12}, \\ a_{02} &= \tfrac{3-c}{8}A_{02} + \tfrac{c}{8}A_{03} + \tfrac{1}{2}A_{12} + \tfrac{1}{8}A_{13}, \\ a_{03} &= \tfrac{1}{2}b_{03} + \tfrac{1}{2}b_{14}, \end{aligned}$$

where $c = \cos\frac{2\pi}{n_0}$ (note that $c = \cos\frac{2\pi}{n_1}$ when constructing a_{31}, a_{32}, a_{13}, and a_{23}). These formulas are derived in § 5.

The points b_{12}, b_{21}, b_{23}, and b_{32} do not affect tangent plane behavior across quad-net boundaries, and may be choosen arbitrarily. Some care should be taken in determining the position of these points so that the resulting surface is free of unwanted undulations or other artifacts. A reasonable construction is:

$$b_{12} = \tfrac{7}{8}A_{12} + \tfrac{1}{8}(A_{21} - A_{11} - A_{22}) + \tfrac{3}{16}(A_{10} + A_{13}) - \tfrac{1}{16}(A_{00} + A_{03}).$$

The remaining Bézier control points are computed by

$$\begin{aligned} b_{11} &= \tfrac{1}{2}a_{10} + \tfrac{1}{2}a_{01}, \\ a_{11} &= \tfrac{1}{2}b_{21} + \tfrac{1}{2}b_{12}, \\ b_{22} &= \tfrac{1}{2}a_{12} + \tfrac{1}{2}a_{21}. \end{aligned}$$

These constructions ensure that the triples $\{b_{01}, a_{00}, b_{10}\}$, $\{a_{01}, b_{11}, a_{10}\}$, $\{b_{12}, a_{11}, b_{21}\}$, and $\{a_{12}, b_{22}, a_{21}\}$ are colinear and share affine ratios. Therefore, the four triangular patches are C^1 along the boundaries internal to a quad-net.

The collection of quartic Bézier triangles constructed in this step are output as the final step in the spline algorithm. Figures 5 and 6 show several control meshes and the corresponding spline surfaces generated by the algorithm.

4.4 Special Cases

The construction just presented generates a smooth spline surface over any control mesh that is topologically a 2-manifold. However, there are certain optimizations that can be implemented to generate patches of lower degree. These special cases arise when n_0 and n_1 equal 3 or 4. In each case, the boundary curves of a quad-net are quadratic rather than cubic. If $n_0 = n_1 = 4$, a single bi-quadratic Bézier patch can be used in place of the four quartic triangles. Otherwise, the four quartic Bézier triangles can be replaced by cubics.

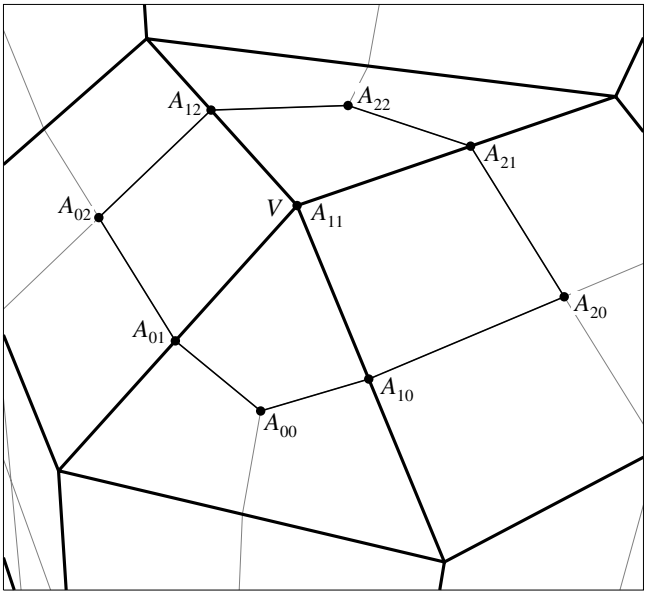
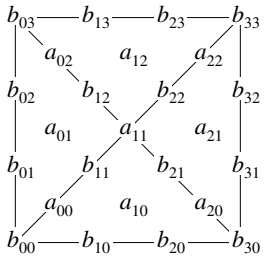


Figure 4: The special case quad-net corresponding to the vertex V.

To take advantage of these optimizations, the special case quad-net shown in Figure 4 corresponding to the V of $\mathcal{M}^1$ is used. The points A_{00}, A_{20}, A_{02}, and A_{22}, are the centroids of the four faces surrounding V. The points A_{10}, A_{01}, A_{12}, and A_{21}, are the midpoints of the four edges incident on V, and the point A_{11} is equivalent to V. This special case quad-net must also know about the integers n_0 and n_1 (equal to the number of sides belonging to the faces surrounding A_{00} and A_{22} respectively).

The four cubic Bézier triangular patches constructed from the special case quad-net are labeled as:



Formulas for the Bézier control net of one of these patches are given. The other three control nets are found by symmetry. The boundary curve is found by

$$\begin{aligned} b_{00} &= A_{00}, \\ b_{01} &= \tfrac{1}{3}A_{00} + \tfrac{2}{3}A_{01}, \\ b_{02} &= \tfrac{2}{3}A_{01} + \tfrac{1}{3}A_{02}, \\ b_{03} &= A_{02}. \end{aligned}$$

Tangent plane continuity is maintained across quad-net boundaries by setting

$$\begin{aligned} a_{00} &= \tfrac{1}{2}b_{10} + \tfrac{1}{2}b_{01}, \\ a_{01} &= \tfrac{1}{6}A_{00} + \tfrac{2-c}{6}A_{01} + \tfrac{1+c}{6}A_{02} + \tfrac{1}{3}A_{11}, \\ a_{02} &= \tfrac{1}{2}b_{02} + \tfrac{1}{2}b_{13}, \end{aligned}$$

where $c = \cos\frac{2\pi}{n_0}$. A smooth join across the internal boundaries is ensured by setting:

$$\begin{aligned} b_{11} &= \tfrac{1}{2}a_{10} + \tfrac{1}{2}a_{01}, \\ a_{11} &= \tfrac{1}{2}b_{21} + \tfrac{1}{2}b_{12}. \end{aligned}$$

If $n_0 = n_1 = 4$, then the special case quad-net is output as the control net of a bi-quadratic tensor product Bézier patch.

5 Smoothness Conditions

The purpose of this section is to derive the constraints imposed on the quad-net construction (§ 4.2), and the formulas for Bézier control points that affect tangent plane behavior along quad-net boundaries (§ 4.3 and 4.4). This section is included for completeness; it is not crucial to understanding the results of this paper.

The purpose of the quad-nets is to characterize the curves and tangent planes along the boundaries of a quadrilateral piece of the spline surface. One such boundary curve is represented by the cubic Bézier curve $[A_{00}, A_{01}, A_{02}, A_{03}]$ constructed in § 4.2. Adjacent quad-nets sharing these points will clearly lead to a continuous (but not necessarily smooth) surface. To see how adjacent quad-nets give rise to surfaces that are tangent plane continous, it must be demonstrated how a quad-net encodes a tangent plane along a boundary.

The tangent plane at a point on a surface can be represented as the span of a pair of vectors. Along a quad-net boundary, one of these vectors is the derivative of the boundary curve written in Bézier form as

$$R = 3[A_{01} - A_{00}, A_{02} - A_{01}, A_{03} - A_{02}].$$

The other vector points inward along the boundary and is defined

$$S = 3[A_{10} - (1-c)A_{00} - cA_{01}, 2A_{12} - A_{01} - A_{02}, A_{13} - A_{03}],$$

where $c = \cos\frac{2\pi}{n_0}$. The tangent plane encoded by the quad-net along the boundary is the span of R and S. Similar expressions hold for the other three edges of a quad-net.

To see that a pair of adjacent quad-nets encode the same tangent plane along a common boundary, consider the pair of quad-nets that share the boundary $[A_{00}, A_{01}, A_{02}, A_{03}]$. Let A_{10}, A_{12}, and A_{13} be the 'first row' of points belonging to the first quad-net adjacent to the common boundary, and let $\hat{A}_{10}$, $\hat{A}_{12}$, and $\hat{A}_{13}$ be the first row of the second quad-net. Clearly, both quad-nets will share the tangent vector R since they share a common boundary curve. By definition

$$\hat{S} = 3[\hat{A}_{10} - (1-c)A_{00} - cA_{01}, 2\hat{A}_{12} - A_{01} - A_{02}, \hat{A}_{13} - A_{03}],$$

is the inward pointing tangent vector of the second quad-net. Adjacent quad-nets will encode the same tangent plane along the common boundary if R, S, and $\hat{S}$ are linearly dependent. This follows by construction, since it is easily verified that Constraints (1-3) are equivalent to the condition

$$S = -\hat{S}.$$

Therefore, the quad-nets constructed in § 4.2 encode identical tangent planes along common boundaries.

Next, it is shown that the triangular patches constructed in § 4.3 interpolate the tangent planes encoded along quad-net boundaries. Let P be the quartic triangular patch constructed to interpolate a quad-net boundary curve. The tangent plane of P along this boundary is the span of the partial derivatives

$$P^u = 4[b_{01} - b_{00}, b_{02} - b_{01}, b_{03} - b_{02}, b_{04} - b_{03}],$$

and

$$P^v = 4[a_{00} - b_{00}, a_{01} - b_{01}, a_{02} - b_{02}, a_{03} - b_{03}]. \qquad (8)$$

The tangent plane of P will interpolate the quad-net tangent plane if R, S, P^u, and P^v are linearly dependent. By construction

$$\begin{aligned} P^u &= R, \quad \text{and} \\ P^v &= \phi R + \psi S, \end{aligned} \qquad (9)$$

where

$$\phi = [\tfrac{1+c}{2}, \tfrac{1}{2}] \quad \text{and} \quad \psi = [\tfrac{1}{2}],$$

are scalar valued functions in Bézier form. Therefore P will interpolate the tangent plane along the boundary of the quad-net. Expanding the right hand side of (9) and equating this result to the right hand side of (8) gives the formulas used to construct points a_{00}, a_{01}, a_{02}, and a_{03}.

5.1 Smoothness in Special Cases

The special case outlined in § 4.4 is similar except the tangent plane encoded by the special case quad-net is the span of the vectors

$$R = 2[A_{01} - A_{00}, A_{02} - A_{01}],$$

and

$$S = 2[A_{10} - (1-c)A_{00} - cA_{01}, A_{11} - A_{01}, A_{12} - A_{02}].$$

Expanding the right hand side of (9) with these definitions of R and S, and equating this result to

$$P^v = 3[a_{00} - b_{00}, a_{01} - b_{01}, a_{02} - b_{02}],$$

gives the formulas used to construct points a_{00}, a_{01}, and a_{02}.

It must also be demonstrated that a special case quad-net of § 4.4 encodes the same boundary curve and tangent plane as a normal quad-net. Let A be a normal quad-net and $\tilde{A}$ be a special case quad-net defined over the same vertex. By construction

$$A_{00} = \tilde{A}_{00}, \quad \text{and} \quad A_{03} = \tilde{A}_{02}.$$

The weights from Theorem 4.1 for the cases $n = 3$ and $n = 4$ are $\{\frac{4}{9}, \frac{4}{9}, \frac{1}{9}\}$ and $\{\frac{5}{12}, \frac{5}{12}, \frac{1}{12}, \frac{1}{12}\}$ respectively. Since these weights are used to construct A_{10}, A_{01}, A_{02}, and A_{13}, it is straightforward to show that in either case

$$\begin{aligned} A_{10} &= \tfrac{1}{3}\tilde{A}_{00} + \tfrac{2}{3}\tilde{A}_{10}, \qquad A_{01} = \tfrac{1}{3}\tilde{A}_{00} + \tfrac{2}{3}\tilde{A}_{01}, \\ A_{02} &= \tfrac{1}{3}\tilde{A}_{02} + \tfrac{2}{3}\tilde{A}_{01}, \qquad A_{13} = \tfrac{1}{3}\tilde{A}_{02} + \tfrac{2}{3}\tilde{A}_{12}, \end{aligned}$$

and

$$A_{12} = \tfrac{1}{6}\tilde{A}_{00} + \tfrac{1}{3}\tilde{A}_{01} + \tfrac{1}{6}\tilde{A}_{02} + \tfrac{1}{3}\tilde{A}_{11}.$$

Substituting these equations (for $A_{00}, \ldots, A_{12}$) into the definition of the tangent vector S for the normal case yields the definition of S for the special case. Therefore, both types of quad-nets encode the same tangent planes and may be used interchangably when n_0 and n_1 are equal to 3 or 4.

6 Conclusions

An algorithm has been presented for constructing a tangent plane smooth spline surface that approximates an irregular control mesh. The spline surface is in general a composite of quartic triangular Bézier patches. In certain special cases, cubic triangular patches may be used in place of the quartics. Over regular regions of the mesh, a bi-quadratic Bézier patch may be used in place of four quartic triangular patches. In fact, the four quartic triangular patches constructed over a regular region represents exactly the same polynomial map as the single bi-quadratic Bézier patch. Although this has not been proved, its plausability is evident since the total degree of a bi-quadratic surface is 4.

The spline algorithm as presented was factored into 3 steps. Each of these steps was a geometric construction that involved taking weighted averages (affine combinations) of points. Therefore, the spline surface is *affine invariant* (i.e., independent of any affine transformation applied to the control mesh). It is not clear that the concatenation of the geometric constructions leads to *convex combinations* in all cases (although the special case constructions of § 4.4 are convex).

Over regular regions of a mesh, the refinement step (§ 4.1) is not needed and will result in more patches being constructed than are actually required. It should be possible to avoid this unnecessary 'splitting' of patches as an optimization.

Appendix

A Treatment of Boundaries

A method of dealing with mesh boundaries in a reasonable way is now presented. The problem is that quad-nets are not defined over boundary vertices of $\mathcal{M}^1$. As a result, the boundary of the spline surface does not approximate the boundary of the $\mathcal{M}^0$ very well. A solution is to modify step 1 (§ 4.1) so that new faces are added to $\mathcal{M}^1$ that correspond to vertices and edges belonging to the boundary of $\mathcal{M}^0$. The following construction has the property that the boundary of the resulting spline surface will be the quadratic B-spline curve corresponding to the boundary vertices of $\mathcal{M}^0$. There are two cases to consider, faces of $\mathcal{M}^1$ corresponding to boundary edges of $\mathcal{M}^0$, and faces of $\mathcal{M}^1$ corresponding to boundary vertices of $\mathcal{M}^0$.

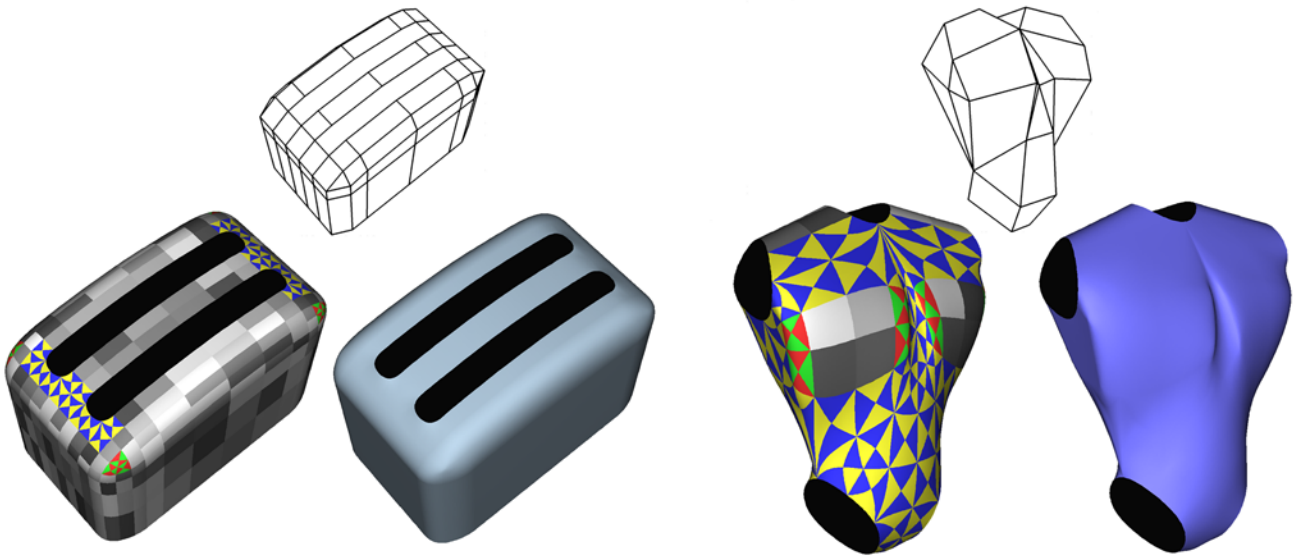


Figure 5: A pair of irregular control meshes and resulting spline surfaces. The patch structure of the spline surfaces are indicated by color: blue and yellow patches are quartic, red and green patches are cubic, and gray patches are bi-quadratic.

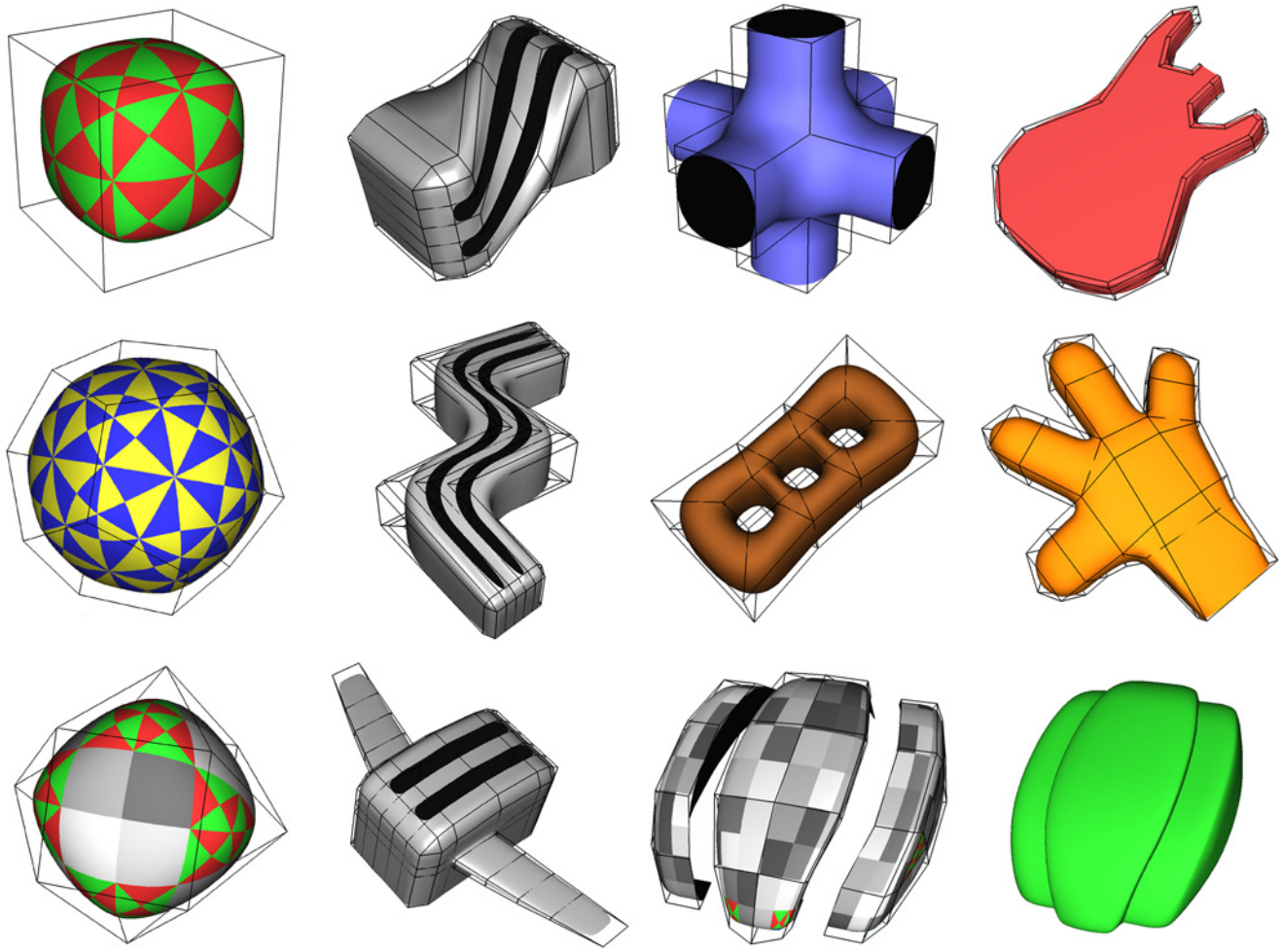


Figure 6: More examples: the color coding of patches is the same as above. The boundaries of meshes are handled by the scheme outlined in Appendix A. This approach may be used to create creases on a surface as illustrated by the two shapes in the lower right hand corner. Disjoint meshes that share boundary geometry will result in a crease.

A.1 Boundary Edges

Let the vertex pair $\{V_0, V_1\}$ be a boundary edge of $\mathcal{M}^0$ belonging to face F. Let P_0 and P_1 be the vertices of $\mathcal{M}^1$ constructed in step 1 corresponding to the vertex-face pairs $\{V_0, F\}$ and $\{V_1, F\}$ respectively. Two new vertices

$$\begin{aligned} Q_0 &= \tfrac{3}{2}V_0 + \tfrac{1}{2}V_1 - P_0, \\ Q_1 &= \tfrac{1}{2}V_0 + \tfrac{3}{2}V_1 - P_1, \end{aligned}$$

and one new face $\{P_0, P_1, Q_1, Q_0\}$ are added to $\mathcal{M}^1$.

A.2 Boundary Vertices

Let V be a vertex on the boundary of $\mathcal{M}^0$. Let k be the number of faces incident on V. Let $P_1, \ldots, P_k$ be the vertices of $\mathcal{M}^1$ corresponding to V constructed in step 1, and let P_0 and P_{k+1} be the vertices found by the boundary edge construction given in Appendix A.1.

When $k = 1$, V is a corner of $\mathcal{M}^0$. By treating this vertex as a discontinuity in the boundary B-spline curve, the spline surface boundary will have a corner. A new face $\{P_0, P_1, P_2, P_3\}$ is added to $\mathcal{M}^1$ where

$$P_3 = 4V - P_0 - P_1 - P_2.$$

When $k > 1$ a new $n = 2k$-sided face $\{P_0, \ldots, P_{n-1}\}$ is added to $\mathcal{M}^1$ where

$$P_i = 2(uQ_0 + (1-u)Q_1) - P_{n-i+1}, \quad i = k+2, \ldots, n-1,$$

with

$$Q_0 = \tfrac{1}{2}P_0 + \tfrac{1}{2}P_1, \quad Q_1 = \tfrac{1}{2}P_k + \tfrac{1}{2}P_{k+1},$$

and

$$u = \tfrac{1}{2}(1 + \cos\tfrac{2\pi i}{n} + \tan\tfrac{\pi}{n}\sin\tfrac{2\pi i}{n}).$$

These constructions, offered without proof, are included because they are of practical value. The boundaries of the spline surfaces illustrated in Figures 5 and 6 were dealt with using this technique.

B Proof of Theorem 4.1

Let $M_k = \beta(\cos\frac{2\pi k}{n} + \tan\frac{\pi}{n}\sin\frac{2\pi k}{n})$. Expand the right hand side of Equation (5) as follows:

$$\begin{aligned} &\tfrac{1}{2}Q_{i-1} + \tfrac{1}{2}Q_{i+1} \\ &= \tfrac{1}{2n}\sum_{j=0}^{n-1} P_j(1 + M_{j-(i-1)}) + \tfrac{1}{2n}\sum_{j=0}^{n-1} P_j(1 + M_{j-(i+1)}), \\ &= \tfrac{1}{2n}\sum_{j=0}^{n-1} P_j(2 + M_{j-i+1} + M_{j-i-1}), \\ &= \tfrac{1}{2n}\sum_{j=0}^{n-1} P_j(2 + 2\cos\tfrac{2\pi}{n}M_{j-i}), \\ &= \tfrac{1}{n}\sum_{j=0}^{n-1} P_j(1 - \cos\tfrac{2\pi}{n}) + \tfrac{1}{n}\sum_{j=0}^{n-1} P_j\cos\tfrac{2\pi}{n}(1 + M_{j-i}), \\ &= (1 - \cos\tfrac{2\pi}{n})O + \cos\tfrac{2\pi}{n}Q_i. \end{aligned}$$

The key step of combining $M_{j-(i-1)} + M_{j-(i+1)}$ to get $2\cos\frac{2\pi}{n}M_{j-i}$ comes about using the well known trigonometric identities:

$$\begin{aligned} \cos\theta + \cos\phi &= 2\cos\tfrac{1}{2}(\theta+\phi)\cos\tfrac{1}{2}(\theta-\phi), \quad \text{and} \\ \sin\theta + \sin\phi &= 2\sin\tfrac{1}{2}(\theta+\phi)\cos\tfrac{1}{2}(\theta-\phi). \end{aligned}$$

Clearly the points Q_i are co-planar since from (5) any Q_i can be found as a linear combination of O, Q_0, and Q_1, and must therefore lie in the plane spanned by these three points.

References

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

[2] H. Chiyokura and F. Kimura. Design of solids with free-form surfaces. In *Proceedings of SIGGRAPH '83*, pages 289–298. 1983.

[3] T. DeRose. *Geometric Continuity: A Parametrization Independent Measure of Continuity for Computer Aided Geometric Design*. PhD thesis, Berkeley, 1985.

[4] D. Doo. A subdivision algorithm for smoothing down irregularly shaped polyhedrons. In *Proceedings on Interactive Techniques in Computer Aided Design*, pages 157–165. Bologna, 1978.

[5] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, third edition, 1993.

[6] T. N. T. Goodman. Closed biquadratic surfaces. *Constructive Approximation*, 7(2):149–160, 1991.

[7] M. Halstead, M. Kass, and T. DeRose. Efficient, fair interpolation using Catmull-Clark surfaces. In *Proceedings of SIGGRAPH '93*, pages 35–44. 1993.

[8] K. Höllig and Harald Mögerle. G-splines. *Computer Aided Geometric Design*, 7:197–207, 1990.

[9] C. Loop. *Generalized B-spline Surfaces of Arbitrary Topological Type*. PhD thesis, University of Washington, 1992.

[10] C. Loop. Smooth low degree polynomial spline surfaces over irregular meshes. Technical Report 48, Apple Computer Inc., Cupertino, CA, January 1993.

[11] C. Loop. A G^1 triangular spline surface of arbitrary topological type. *Computer Aided Geometric Design*, 1994. to appear.

[12] C. Loop and T. DeRose. Generalized B-spline surfaces of arbitrary topology. In *Proceedings of SIGGRAPH '90*, pages 347–356. 1990.

[13] J. Peters. C^1 free-form surface splines. Technical Report CSD-TR-93-019, Dept. of Comp. Sci., Purdue University, W-Lafayette, IN, March 1993.

[14] J. Peters. Smooth free-form surfaces over irregular meshes generalizing quadratic splines. *Computer Aided Geometric Design*, 10:347–361, 1993.

[15] J. Peters. Constructing C^1 surfaces of arbitrary topology using biquadratic and bicubic splines. In N. Sapidis, editor, *Designing Fair Curves and Surfaces*. 1994. to appear.

[16] U. Reif. Biquadratic G-spline surfaces. Technical report, Mathematisches Institut A, Universität Stuttgart, Pfaffenwaldring 57, D-7000 Stuttgart 80, Germany, 1993.

[17] M. Sabin. 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.

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