1993

Smooth Splines Over Irregular Meshes Built From Few Polynomial Pieces of Low Degree

Jörg Peters

Report Number:
93-019

SMOOTH SPLINES OVER IRREGULAR MESHES BUILT FROM FEW POLYNOMIAL PIECES OF LOW DEGREE

Jorg Peters

CSD-TR-93-019
March 1993
(Revised April 1993)

Smooth splines over irregular meshes built from few polynomial pieces of low degree

by

Jörg Peters†
jorg@cs.purdue.edu

Key words: C^1 surface, corner cutting, box splines, C^1 spline mesh, blending

Running title: Free-form spline meshes

Version: April 15 93 **Date printed:** April 21, 1993

Submitted to: SIAM J Num Anal

Abstract

A 3D mesh of points outlining a polytope of arbitrary topological structure serves as the control mesh of a spline surface with pieces of degree two and three. The user has a choice between three-sided or four-sided pieces or a combination of the two to parametrize the surface. If the surface is constructed entirely from bicubics and the input polytope has e edges, then the surface consists of $2e$ pieces. Where the mesh of points is regular a quadratic spline surface is generated. Irregular input meshes with non quadrilateral mesh cells and more or fewer than four cells meeting at a point are allowed and generate splines that generalize the quadratic C^1 spline construction by averaging. The algorithm can model bivariate open or closed surfaces of arbitrary topological structure.

† Department of Computer Science, Purdue University, W-Lafayette IN 47907-1398
Supported by NSF grant CCR-9211322

1. Introduction

Splines assembled from B-splines are widely used to represent surfaces. They combine a low degree polynomial or rational representation of maximal smoothness with a geometrically intuitive variation of the surface in terms of the coefficients: by connecting the coefficients one obtains a mesh that roughly outlines the surface. Repeated refinement of this mesh by knot insertion results in a sequence of meshes whose points are averages of the preceding and whose limit is the surface itself. In addition to an elegant algebraic definition this yields an alternative geometric, procedural characterization of the splines useful for establishing many shape properties of spline surfaces. But the tensor product B-spline representation has a major shortcoming. It cannot model certain real world objects without singularity, because each point in the interior of the B-spline mesh must be *regular*, that is surrounded by exactly four quadrilateral mesh cells. This makes it impossible to choose for example the boundary mesh of a cube as input and in fact restricts the topological structure of the objects that can be modeled by the splines. Even if the object to be modeled can be described as a deformation of the plane, it may be more natural to have three or five quadrilaterals join at a point or to use non quadrilateral cells to model a feature. Using trimmed NURBS (non uniform rational B-splines) does not solve this problem since the trimming destroys one of the chief advantages of the B-spline representation, its built in smoothness. One ends up with the tricky task of smoothly joining the trimmed pieces. The goal is therefore to devise algorithms that remove the regularity restrictions from the input mesh and yield a unified approach to surface modeling. The approach should reduce to the B-spline paradigm wherever the mesh is regular and have the following additional properties.

- *smooth free-form modeling capability.* There are no restrictions on the number of cells meeting at a mesh point or the number of edges to a mesh cell. Mesh cells need not be planar.
- *local smoothness preserving editability.* The component functions of the spline surface form a vector space of smooth functions. In order to add and subtract the functions and locally edit the geometry, it suffices to add and subtract the mesh points locally.
- *low degree parametrization.* The surface is parametrized by low degree polynomial patches. The representation can be extended to rational patches.
- *simple interpolation.* It is possible to interpolate the input mesh points and normals without solving a system of constraints.
- *construction by averaging.* The coefficients of the parametrization can be obtained by applying averaging masks to the input mesh. Thus the algorithm is local and can be interpreted as a rule for cutting an input polytope such that the limit polytope is the spline surface.
- *convex hull property.* The surface lies locally and globally in the convex hull of the input mesh.
- *intuitive shape parameters.* The averaging or cutting process is geometrically intuitive. Smaller cuts result in a surface that follows the input mesh more closely and changes the normal direction more rapidly across the boundary.
- *taut interpolation of a wire frame as a limit case.* Cuts of zero depth result in a singular parametrization at the mesh points analogous to singularities of a quadratic

spline with repeated knots. The C^1 surface degenerates into a C^0 surface that interpolates the edges of the input mesh and remains taut, e.g. planar when the mesh cell is planar.

In summary, one would like an algorithm that departs as little as possible from the NURB standard and combines the intuitive cutting paradigm with a low degree parametrization.

The algorithm described in this paper generalizes the quadratic C^1 spline paradigm for both tensor-product B-splines and four-direction box splines to generate a surface parametrized by four-sided or three-sided Bernstein-Bézier patches. Both the Bernstein-Bézier form and box splines are standard tools of geometric modeling. [Boehm, Farin, Kahmann '84] and [de Boor, Höllig, Riemenschneider 92] are good references. All the above properties are satisfied, except if a vertex has $2n > 4$ neighbors and is surrounded entirely by four-sided patches. Then the surface is only guaranteed to minimize the residual of the smoothness constraints. For surfaces built from three-sided patches or a combination of three-sided and four-sided patches, the surface can always be guaranteed to be tangent plane continuous.

The present method generates a spline space over irregular meshes by local averaging as do the algorithms in [Peters 92a], [Peters 92b] and [Loop '93]. The first of the three earlier algorithms contributes the idea of mesh refinement to separate irregular vertices, the second adds the quadratic spline representation and the third uses the idea of forcing all boundary curves to be quadratic. The main improvements of the present over the earlier algorithms are as follows.

- Considerably fewer surface pieces are generated. While the earlier schemes generate 16 four-sided or 48 three-sided patches for each regular mesh point, the current algorithm uses only one fourth that number.
- The surface is guaranteed to lie in the convex hull of the control mesh for the full range of blend ratios.
- Parametrization with three-sided and four-sided patches is treated uniformly and can be mixed.

The algorithms in [Sabin '83], [Goodman '88] and [Höllig, Mögerle '89], also define spline spaces. However, these constructions are less localized and need to solve large irregularly sparse systems of equations to match data. This makes it more difficult to reason about the shape of the resulting surface. Algorithms for generalized subdivision ([Sabin '76], [Doo '78], [Catmull and Clark '78], [Loop '87], [Dyn, Levin and Liu '92], etc.) differ in that they do not provide an explicit parametrization for the irregular mesh regions. This not only makes it tricky to establish elementary properties like tangent plane continuity of the limit surface, but is also a major obstacle for integrating these techniques with other computer aided design representations. The work on reparametrization and geometric smoothness is the background for the current algorithm (see e.g. [Gregory '90] for a survey). In particular, the construction based on three-sided patches can be viewed as a special case of [Peters '90] with a well-chosen mesh of quadratic boundary curves as input. The construction based entirely on four-sided patches generalizes the construction of [van Wijk '86, Section 3.5] to the case where $n > 4$ patches meeting at a mesh point. The algorithm of this paper differs from schemes like [Sarraga '87],[Hahn '89] in that no constraint systems have to be solved to enforce patch to patch smoothness. B-patches ([Seidel '91], [Dahmen, Micchelli,

Seidel '9x]) and S-patches [Loop, DeRose '90] use a special representation of the surface pieces, while the present algorithm uses either standard tensor-product or triangular pieces in Bernstein-Bézier form.

The algorithm is detailed in Section 2. Section 3 establishes the continuity and vector space properties of the surfaces generated by the algorithm. Section 4 establishes the shape properties of the surfaces generated by the algorithm. Section 5 summarizes the findings and the Appendix illustrates the algorithm.

2. An algorithm for refining an irregular mesh of points into a C^1 surface

The algorithm has six simple steps. The output of steps 1-3 is a mesh of quadrilateral cells such that each original vertex is surrounded by vertices of degree four. Steps 4,5 and 6 generate the surface parametrization and depend on the user's choice of three-sided or four-sided or mixed patch representation.

1 Mesh refinement.
2 Edge cutting.
3 Quadratic meshing.
4 Quadratic patching.
5 Degree raising.
6 Twist adjustment.

The *input* is any mesh of points such that at most two cells abut along any edge. The mesh cells need not be planar, and there is no constraint on the number of cells meeting at a vertex. However, if a cell has more than four sides, then the valency of its vertices, i.e. the number of neighbors, is restricted to be 3 or 4; this can be enforced for example by triangulating the cell. The mesh may model a bivariate open or closed surface of arbitrary topological structure. For each vertex of a cell, there are two scalar weights $0 < a_i < 1$, $i = 1, 2$, called *blend ratios.* Geometrically, smaller ratios results in a surface that follows the input mesh more closely and changes the normal direction more rapidly close to the mesh edges (cf. A.3 of the Appendix). The default is $a_i := 1/2$. Algebraically, blend ratios play the role of relative knot spacings. In particular $a_i = 0$ implies loss of smoothness. The blend ratios of each cell may be modified independently. The *output* is a surface that follows the outline of the input mesh and consists, depending on the users choice of surface representation, of either

(a) no more than $8e$ quadratic or cubic, three-sided patches that form a C^1 surface, where e is the number of edges of the input mesh; or
(b) no more than $2e$ biquadratic or bicubic, four-sided patches that form a C^1 surface except possibly at vertices and faces with $2n > 4$ neighbors (see Theorem 3.5); or
(c) a combination of biquadratic, four-sided and cubic, three-sided patches.

The construction with 4-sided patches is least recommended since neither tangent plane continuity nor the convex hull property can be guaranteed for all configurations of the input mesh. All indices are interpreted modulo n. All coefficients V, C, P, etc. are vectors.

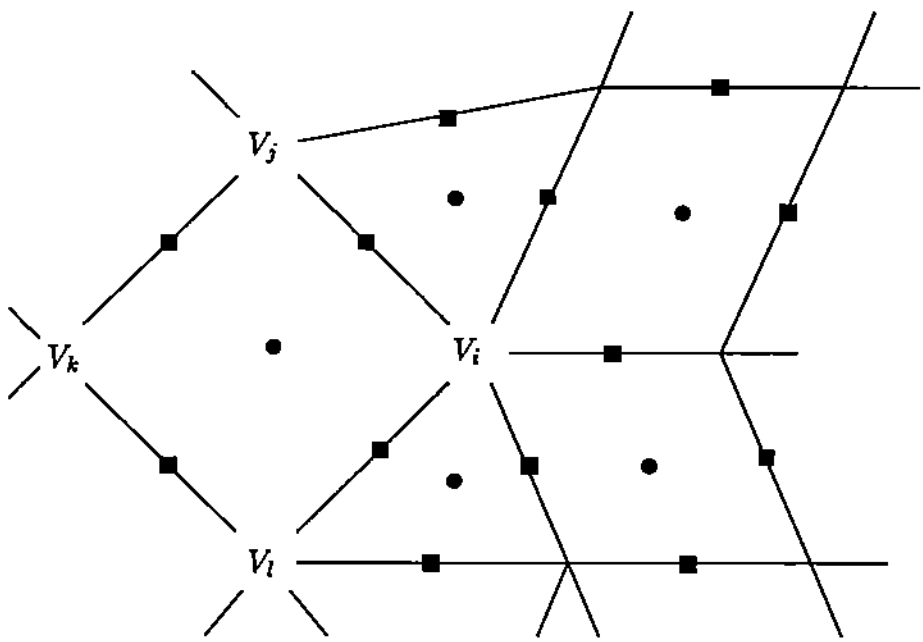
Step 1: Mesh refinement. For each edge with vertices V_i and V_j, create an edge vertex

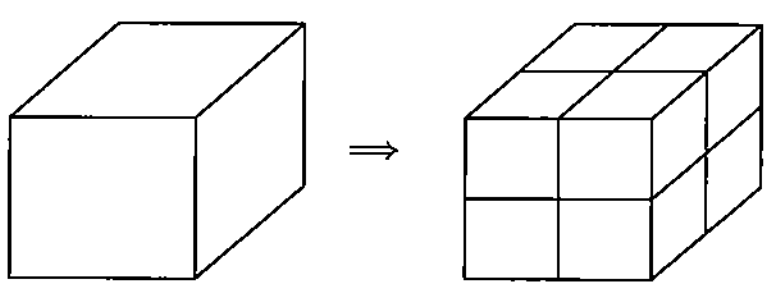
$$\blacksquare := (V_i + V_j)/2$$

and insert it between V_i and V_j. For each cell c with vertices $V_1, V_2, \ldots, V_m$, create a cell vertex

$$\bullet := \frac{1}{m} \sum V_i$$

and connect it to all edge vertices of the cell. This creates a refined mesh of subcells.





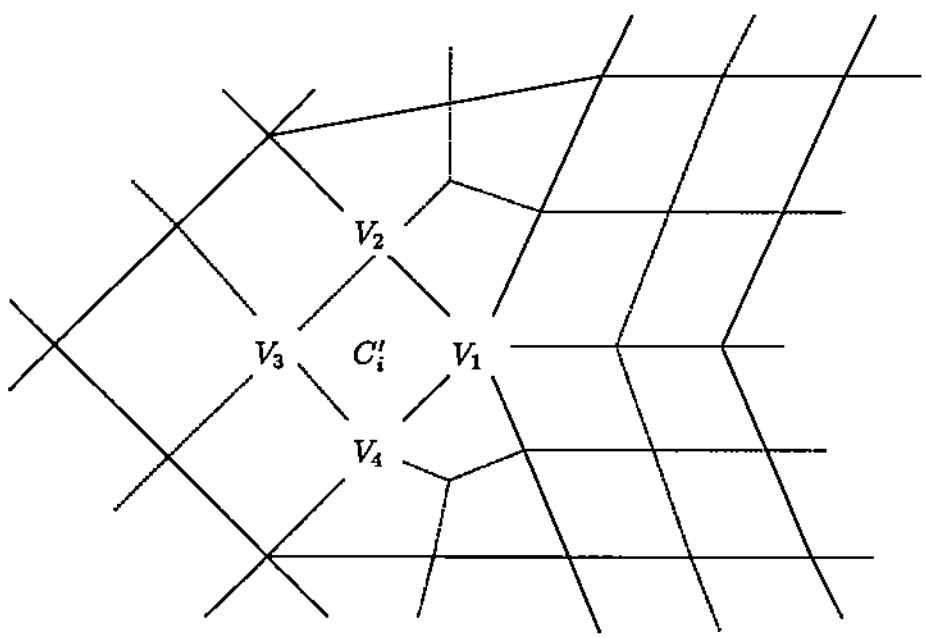
Step 2: Edge cutting. For each subcell c_i label the vertices V_1, V_2, V_3, V_4 in order starting with the input mesh point, and create a preliminary cell center C_i' subject to the blend ratios $0 < a_{i1}, a_{i2} < 1$ as the average

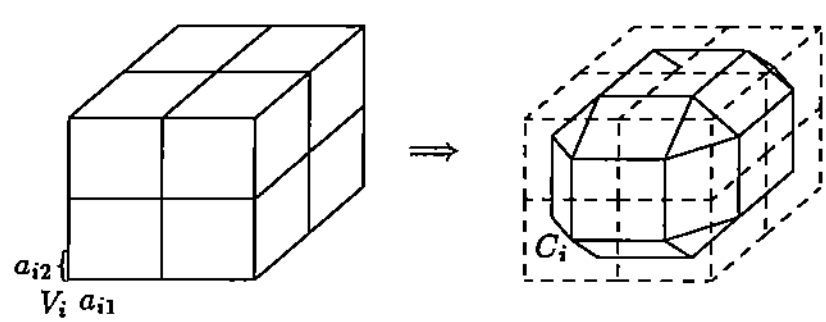
$$C_i' := (1-a_{i1})(1-a_{i2})V_1 + (1-a_{i1})a_{i2}V_2 + a_{i1}a_{i2}V_3 + (1-a_{i2})a_{i1}V_4,$$

For each vertex V surrounded by less than five cells, the cell center is $C_i = C_i'$. For each vertex V surrounded in clockwise order by more than four cells $c_1, c_2, \ldots, c_n$, the cell center corresponding to c_i is

$$C_i := \bar{V} + \frac{2\omega_n}{n}\sum_j \cos(\frac{2\pi}{n}j)C_{i+j}' \tag{2.1}$$

where $\bar{V} := \frac{1}{n}\sum C_i'$, and $\omega_n^{-1} := -\cos(\frac{n}{2}\frac{2\pi}{n}) - \cos(\frac{n+1}{2}\frac{2\pi}{n})$. If V is to be interpolated set $\bar{V} = V$ instead.



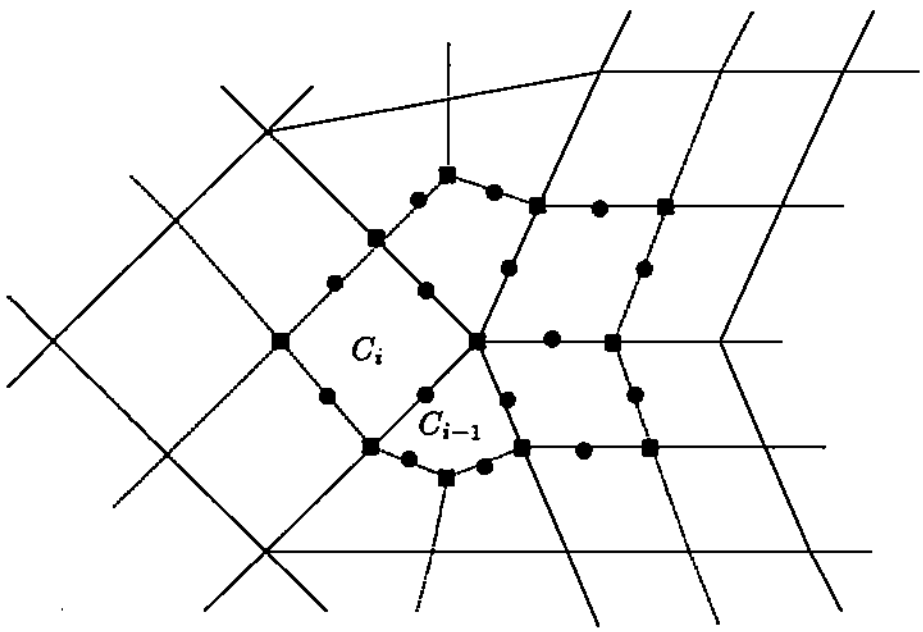


Step 3: Quadratic Meshing. For each vertex V surrounded in order by cell centers $C_1, C_2, \ldots, C_n$, compute the final location

$$\blacksquare := \bar{V} := \frac{1}{n} \sum C_i.$$

For each edge V, V_i separating two subcells with centers C_{i-1}, C_i, create an edge coefficient

$$\bullet \ := A_i := (C_i + C_{i-1})/2.$$

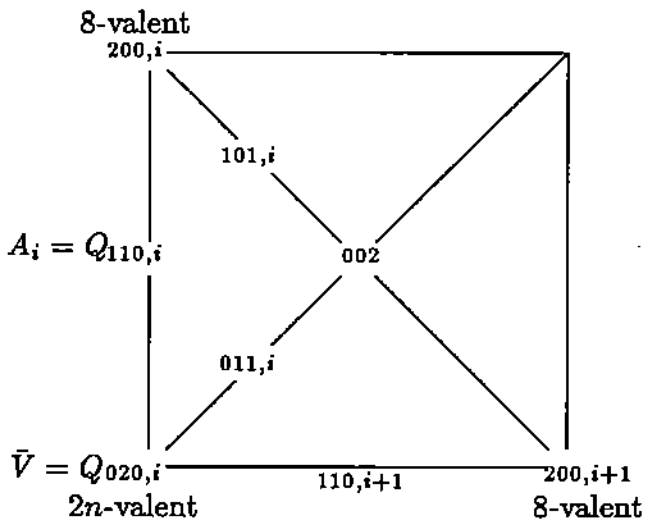


Result: a mesh of quadrilateral cells. Associated with each edge are now three coefficients that may be interpreted as the Bernstein-Bézier coefficients of a quadratic curve segment. This yields a mesh of quadrilaterals with quadratic boundary curves and such that each coefficient $\blacksquare$ lies in the same plane as the surrounding A_i's.

Three-sided Patches.

Step 4T: Quadratic Patching. Each quadrilateral subcell is covered by four triangular patches. The interior coefficients are computed by averaging:

$$Q_{101,i} = Q_{011,i+1} = \frac{Q_{110,i} + Q_{110,i+1}}{2}, \qquad Q_{002} = \frac{Q_{101,0} + Q_{101,2}}{2}.$$



Step 5T: Degree raising. Raise the degree to obtain the coefficients of the triangular cubic patch (*ed coefficients need not be computed)

$$\begin{array}{ccccc}
P_{300} & & & Q_{200} & \\
 & P_{201} & & \frac{Q_{200}+2Q_{101}}{3} & \\
P_{210} & & P^*_{102} & \frac{Q_{200}+2Q_{110}}{3} & \frac{Q_{002}+2Q_{101}}{3} \\
 & P^*_{111} & & \frac{Q_{110}+Q_{101}+Q_{011}}{3} & \\
P_{120} & & P^*_{012} & \frac{Q_{020}+2Q_{110}}{3} & \frac{Q_{002}+2Q_{011}}{3} \\
 & P_{021} & & \frac{Q_{020}+2Q_{011}}{3} & \\
P_{030} & & & Q_{020} &
\end{array} \qquad P^*_{003} = \qquad Q_{002}$$

Step 6T: Twist Adjustment. At a vertex $V = P_{030}$ surrounded by n quadrilateral cells, set $c := \cos(\frac{2\pi}{n})$ and compute with C_i the center of the ith cell

$$\begin{aligned}
P_{111,i} &= [(2-c)Q_{110,i} + (1+c)Q_{200,i} + 2C_i + Q_{020,i}]/6 \qquad (2.2)\\
P_{102,i} &= (P_{111,i} + P_{111,i+1})/2,\\
P_{003,i} &= (P_{102,0} + P_{102,2})/2.
\end{aligned}$$

This and the previous step are not necessary if $n = 4$.

Four-sided Patches.

Step 4R: Quadratic Patching. Associated with each subcell are four vertex, four edge and one center coefficient that may be interpreted as the Bernstein-Bézier coefficients of a biquadratic patch. For example, $Q_{00} = \bar{V}$, $Q_{01} = A_i$ and $Q_{11} = C_i$.

Step 5R: Degree raising. For each subcell, the degree of the associated biquadratic patch degree is raised to bicubic. That is,

$$\begin{bmatrix} Q_{00} & Q_{10} & Q_{20} \\ Q_{01} & Q_{11} & Q_{21} \\ Q_{02} & Q_{12} & Q_{22} \end{bmatrix} \Longrightarrow \begin{bmatrix} P_{00} & P_{10} & P_{20} & P_{30} \\ P_{01} & P_{11}^* & P_{21} & P_{31} \\ P_{02} & P_{12} & P_{22}^* & P_{32} \\ P_{03} & P_{13} & P_{23} & P_{33} \end{bmatrix} =$$

$$= \begin{bmatrix} Q_{00} & \frac{2Q_{10}+Q_{00}}{3} & \frac{2Q_{10}+Q_{20}}{3} & Q_{20} \\ \frac{2Q_{01}+Q_{00}}{3} & \frac{4Q_{11}+2(Q_{01}+Q_{10})+Q_{00}}{9} & \frac{4Q_{11}+2(Q_{21}+Q_{10})+Q_{20}}{9} & \frac{2Q_{21}+Q_{20}}{3} \\ \frac{2Q_{01}+Q_{02}}{3} & \frac{4Q_{11}+2(Q_{01}+Q_{12})+Q_{02}}{9} & \frac{4Q_{11}+2(Q_{21}+Q_{12})+Q_{22}}{9} & \frac{2Q_{21}+Q_{22}}{3} \\ Q_{02} & \frac{2Q_{12}+Q_{02}}{3} & \frac{2Q_{12}+Q_{22}}{3} & Q_{22} \end{bmatrix}$$

Step 6R: Twist Adjustment. For each vertex V surrounded clockwise by patches p_i, $i = 1..n$, let $V = P_{00,i}$ and $P_{01,i-1} = P_{10,i}$ and define

$$E_i := P_{10,i} + \frac{c}{3}(P_{30,i} - P_{20,i}), \quad c := \cos(\frac{2\pi}{n}).$$

Then we recompute

$$P_{11,i-1} = \begin{cases} -\sum_{j=1}^{n}(-1)^j E_{i+j} & \text{if } n \text{ is odd} \\ \frac{-2}{n}\sum_{j=1}^{n}(n-j)(-1)^j E_{i+j} & \text{if } n \text{ is even.} \end{cases} \tag{2.3}$$

This and the previous step are not necessary if $n = 4$.

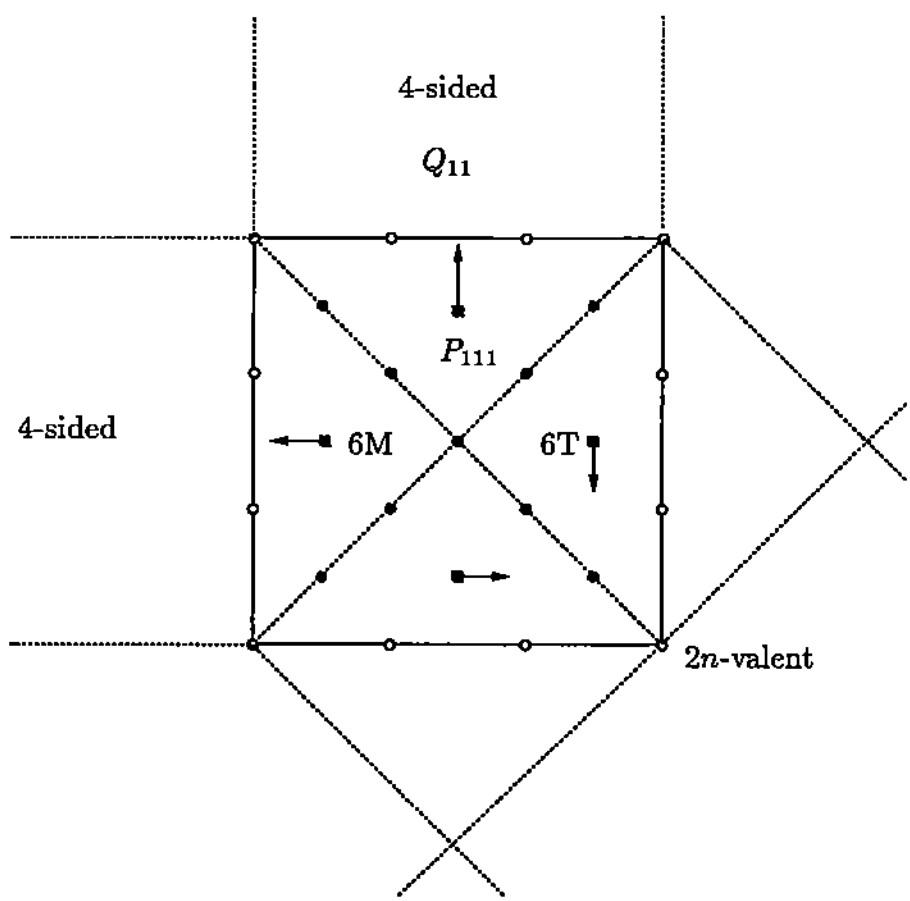
Three-sided cubics joining biquadratic patches.

To join a three-sided patch p smoothly to a biquadratic four-sided patch q, denote as Q_{00}, Q_{10}, and Q_{20} the coefficients of the common quadratic boundary and as Q_{11} the center coefficient of the biquadratic patch. Patch p is constructed by steps $4T$ and $5T$ as usual but the perturbation step determines the center coefficient as

Step 6M: Twist Adjustment.

$$P_{111} = Q_{10} + \frac{1}{3}(\frac{Q_{20} + Q_{02}}{2} - Q_{11}).$$

The join allows filling n-sided holes in a biquadratic tensor-product spline surface using three-sided cubic patches (cf. A.1 of the Appendix). A hole is first divided into n quadrilateral cells. Then each cell is covered by four three-sided, cubic patches. Since the center coefficients of the cubic patches attached to the $2n$-valent central vertex are not edge-adjacent to the biquadratic patches, they are adjusted as in Step 6T.



3. Continuity and vector space properties

This section shows that splines generated by the surface form a smooth vector space. Smoothness, oriented tangent plane continuity is characterized as the agreement of the derivatives of two maps p and q from $\mathbb{R}^2$ to $\mathbb{R}^n$ after reparametrization by a map φ from $\mathbb{R}^2$ to $\mathbb{R}^2$ that connects the domains Ω_p and Ω_q of p and q:

$$p = q \circ \varphi \quad \text{and} \quad D_1 p = D_1(q \circ \varphi) \quad \text{along } E_p$$

where $\varphi(E_p) = E_q$, E_p and E_q are edges of Ω_p and Ω_q respectively. D_1 denotes differentiation in the direction perpendicular to E_p and φ maps interior points of Ω_q to exterior points of Ω_p thus avoiding cusps. We prove oriented tangent plane continuity first for the construction with three-sided patches, then for the construction with four-sided patches and finally for the mixed construction. We prepare the result with two lemmas. The first records the mesh structure after the refinement step.

(3.1) Lemma. *Steps 1–3 generate a mesh of quadrilaterals bounded by quadratic curves and such that at least one vertex of every edge has exactly 4 neighbors.*

(3.2) Lemma. *The coefficients A_i constructed in Step 3 at V satisfy*

$$A_{i-1} - 2cA_i + A_{i+1} = 2(1-c)V, \quad i = 1..n, \quad c := \cos(\frac{2\pi}{n}).$$

Proof For simplicity we center the coordinate system at $V = \frac{1}{n}\sum C_i = 0$. For $n = 3$ and $n = 4$, the statement follows directly from the fact that $A_i := \frac{C_{i-1}+C_i}{2}$. For $n > 4$, Step 2 enforces

$$\begin{aligned} &C_{i-1} - 2cC_i + C_{i+1} \\ &= \frac{2\omega_n}{n}\sum_{i=1}^{n} C_i\big(\cos(\frac{2\pi}{n}(i-1)) + \cos(\frac{2\pi}{n}(i+1)) - 2\cos(\frac{2\pi}{n})\cos(\frac{2\pi}{n}i)\big) = 0 \end{aligned}$$

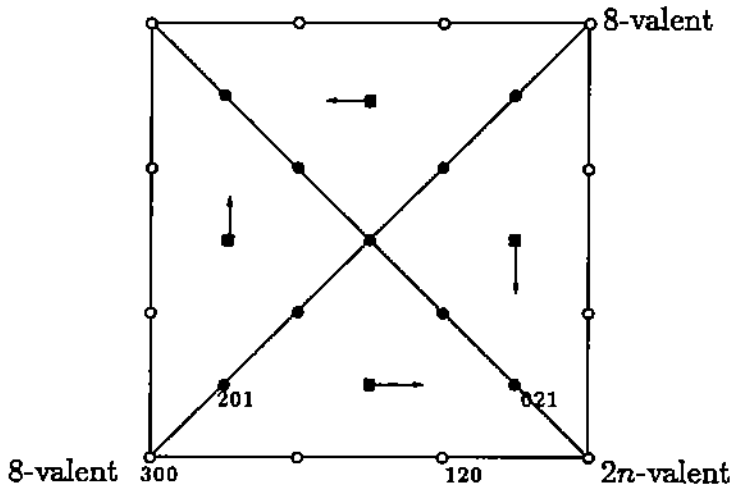
and hence the same holds for the averages A_i. ■

By Lemma 3.1 and Step 4T, each three-sided patch has one vertex with eight neighbors and one vertex with four neighbors. Therefore, the central coefficient P_{111} is associated with at most one vertex that is neither 8-valent nor 4-valent. The arrows in the diagram below indicate this association. (o labels the degree-raised boundaries generated by Step 5T, • the averaged coefficients of the interior boundaries and ■, represents the central coefficient P_{111} constructed in Step 6T.)

(3.3) Theorem. *Steps 1–3 and 4T–6T of the algorithm generate a quadratic-cubic C^1 surface.*

Proof Step 6T enforces

$$D_1 p_{j-1\,|e} = D_1(p_j \circ \psi)_{|e}, \text{ where } \psi := \mathrm{id} + t_1 \begin{bmatrix} 0 \\ 1 - ct_2 \end{bmatrix} \quad \text{and } c := \cos(\frac{2\pi}{n})$$



for adjacent patches p_i and p_{i-1}. Let Q_{200}, Q_{110} and Q_{020} be the Bernstein-Bézier coefficients of the common quadratic boundary curve; Q_{200} corresponds to $t_2 = 0$ where 4 quadrilaterals meet and Q_{020} corresponds to $t_2 = 1$ where n quadrilaterals meet. Then the constraints on the coefficients of the cubic patches are

$$P_{201,j-1} - P_{300,j-1} + P_{201,j} - P_{300,j} = \frac{2}{3}(Q_{110} - Q_{200})$$
$$P_{111,j-1} - P_{210,j-1} + P_{111,j} - P_{210,j} = \frac{1}{3}\big(Q_{020} - Q_{110} + (1-c)(Q_{110} - Q_{200})\big)$$
$$P_{021,j-1} - P_{120,j-1} + P_{021,j} - P_{120,j} = \frac{2}{3}(1-c)(Q_{020} - Q_{110})$$

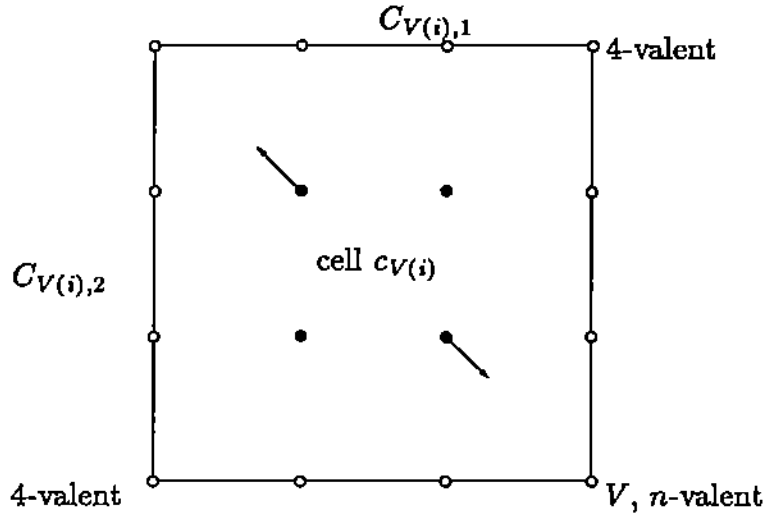
The first and the third equation hold by Lemma 3.2, the second is checked by substitution. ■

(3.4) Corollary. *Steps 4T-6T cover cells with no irregular mesh points with a quadratic box spline.*

Proof When all cell vertices are regular mesh points, then $c = 0$ in Step 6T and the construction leaves the patch unchanged as a quadratic after Step 4T. By Theorem 3.3, the surface is C^1 and the connecting map the identity. Therefore the patches join parametrically C^1 and must be identical with the quadratic box spline (cf. [de Boor, Höllig, Riemenschneider 92]). ■

For $i = 1..n$, let c_i be the subcells that surround V and $C_{i,1}$ and $C_{i,2}$ the center coefficients of the two subcells that share an edge with c_i, but do not have V as a vertex. Define

$$E(V) := \frac{-1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{2}(-1)^{i+j}C_{i,j}.$$



(3.5) Theorem. *Steps 1–3 and 4R–6R of the algorithm generate a biquadratic-bicubic C^1 surface except at a vertex V with $2n > 4$ neighbors and $E(V) \neq 0$. In the latter case the residual of the C^1 constraints is minimized by the construction.*

Proof Unless $m = 2n > 4$ and $E(V) \neq 0$, the construction enforces

$$D_1 p_{j-1|e} = D_1(p_j \circ \psi)_{|e}, \text{ where } \psi := \mathrm{id} + t_1(1-t_2)^2 \begin{bmatrix} 0 \\ 2c \end{bmatrix} \quad \text{and } c := \cos(\frac{2\pi}{n}).$$

Let Q_{20}, Q_{11} and Q_{02} be the Bernstein-Bézier coefficients of the common quadratic boundary curve of p_{j-1} and p_j; Q_{20} corresponds to $t_2 = 0$ where 4 quadrilaterals meet and Q_{02} corresponds to $t_2 = 1$ where n quadrilaterals meet. The equation is verified by writing out the conditions in Bernstein-Bézier form:

$$\begin{aligned} P_{30,i} - P_{31,i} &= P_{13,i-1} - P_{03,i-1} \\ P_{20,i} - P_{21,i} &= P_{12,i-1} - P_{02,i-1} \\ \frac{4c}{9}(Q_{20,i} - Q_{10,i}) + P_{10,i} - P_{11,i} &= P_{11,i-1} - P_{01,i-1} \\ \frac{4c}{3}(Q_{10,i} - Q_{00,i}) + P_{00,i} - P_{01,i} &= P_{10,i-1} - P_{00,i-1} \end{aligned}$$

The first, second and fourth constraint are satisfied by the choice of ■ in Step 2. With $P_{30,i} - P_{20,i} = \frac{2}{3}(Q_{20,i} - Q_{10,i})$, and $P_{10,i} = P_{01,i-1}$, the third constraint is

$$\frac{P_{11,i} + P_{11,i-1}}{2} = E_i := P_{10,i} + \frac{c}{3}(P_{30,i} - P_{20,i}).$$

Choosing $P_{11,i}$ according to (2.3) enforces the constraint if n is odd:

$$P_{11,i-1} + P_{11,i} = -\sum_{j=1}^{n}(-1)^j(E_{i+j-1} + E_{i+j}) = 2E_i.$$

If n is even, then

$$P_{11,i-1} + P_{11,i} = \frac{-2}{n}\sum_{j=1}^{n}(n-j)(-1)^j(E_{i+j-1} + E_{i+j}) = \frac{-2}{n}(\sum_{j=1}^{n}(-1)^j E_j - nE_i)$$
$$= 2E_i - 2E/n.$$

That is, the total residual $2E$ is equally distributed over the patch boundaries emanating from the vertex. ■

(3.6) Corollary. *Steps 4R-6R cover cells with no irregular mesh points with a biquadratic (B-)spline.*

Proof If both vertices of an edge e have valency 4, then the transversal derivatives of the patches p_i and p_{i-1} that share the edge are collinear and of equal size. Hence the patches are joined parametrically C^1. ■

(3.7) Theorem. *Steps 1–3 and 4T,5T and 6M of the algorithm generate a biquadratic-cubic C^1 surface.*

Proof Due to Theorems 3.3 and 3.5, we need only consider the transition between q, a biquadratic patch and p a cubic patch. Then

$$D_1 p_{|e} = D_1(q \circ \psi)_{|e}, \text{ where } \psi := \mathrm{id} + t_1 \begin{bmatrix} -\frac{1}{2} \\ \frac{1}{2} \end{bmatrix}.$$

Let Q_{20}, Q_{11} and Q_{02} be the Bernstein-Bézier coefficients of the common quadratic boundary curve; Q_{20} corresponds to $t_2 = 0$ and Q_{02} corresponds to $t_2 = 1$. Then the constraints on the Bernstein-Bézier coefficients are

$$Q_{10} - Q_{00} = Q_{01} - Q_{00} + 3(P_{210} - P_{300}),$$
$$\frac{Q_{20} - Q_{00}}{2} = Q_{11} - Q_{10} + 3(P_{111} - P_{210}),$$
$$Q_{20} - Q_{10} = Q_{21} - Q_{20} + 3(P_{021} - P_{120}).$$

The first and the third constraint are satisfied by the quadratic construction. Since $P_{210} = \frac{Q_{20}+2Q_{10}}{3}$, the second constraint is

$$P_{111} = \frac{1}{3}(Q_{20} + 2Q_{10} + Q_{10} - Q_{11} + \frac{Q_{00} - Q_{20}}{2}) = Q_{10} + \frac{1}{3}(\frac{Q_{20} + Q_{02}}{2} - Q_{11}).$$

■

(3.8) Theorem. *C^1 surfaces generated from input meshes with the same connectivity, choice of three-sided and four-sided patches, and blend ratio for each subcell form a vector space.*

Proof Two surfaces generated from input meshes with the same connectivity have a natural 1-1 and onto correspondence of patches. Consider two smoothly abutting patches

p_i and q_i, $i = 1, 2$ of the ith surface. The connecting map depends only on the connectivity of the mesh via c. Identifying the open neighborhood of the edges $E_{p,1}$ and $E_{p,2}$ as E_p, there is a single connecting map φ such that

$$p_i = (q_i \circ \varphi) \quad \text{and} \quad D_1 p_i = D_1(q_i \circ \varphi) \quad \text{along } E_p.$$

Consequently, if $p_0 := \beta_1 p_1 + \beta_2 p_2$ and $q_0 := \beta_1 q_1 + \beta_2 q_2$, then

$$p_0 = (q_0 \circ \varphi) \quad \text{and} \quad D_1 p_0 = D_1(\beta_1 p_1 + \beta_2 p_2) = \beta_1 q_1 \circ \varphi + \beta_2 q_2 \circ \varphi = D_1(q_0 \circ \varphi)$$

along E_p as claimed. ■

Spreading the $x-y$ coordinates in the plane and choosing all z coordinates of the mesh equal to zero except for one, results in familiar B-spline like humps. Adding these humps yields the constant function 1 since the C_i and hence all coefficients have z component 1 and the surface is continuous.

4. Shape properties of the resulting surface

This section establishes the convex hull property of the spline space induced by the algorithm. Additionally it is shown that surfaces are flat in the neighborhood of the image of a mesh point if and only if the mesh is locally flat. The edges of the input mesh are interpolated and thus the outlines of the input polytope recaptured when the blend ratios are zero. Finally, if the blend ratios of an input mesh cell agree, then the centroid is interpolated.

The following Lemma contains the essence of the proof of the convex hull property.

(4.1) Lemma. *Step 2 forces the cell centers C_i into the same plane such that $A_i := \frac{C_i+C_{i-1}}{2}$ lie in the convex hull of the original points.*

Proof Choosing the local coordinate system such that $\bar{V} := \frac{1}{n}\sum_j C'_j = 0$, one wants to find appropriate $C_i \neq 0$ in the null space of the constraint matrix corresponding to the smoothness constraint $C_{i-1} - 2\cos(\frac{2\pi}{n})C_i + C_{i+1} = 0$. This null space is spanned by (two of) the cyclical permutations of the vectors $(\cos\frac{2\pi}{n}\ell)_{\ell=1..n}$. Using these vectors symmetrically and applying a single weight ω_n, a natural solution to the coplanarity problem is

$$C_i := \frac{2\omega_n}{n}\sum_j \cos(\frac{2\pi}{n}j)C'_{i+j}.$$

The coplanarity itself follows from Lemma 3.2. Define

$$a_{i,j} := \frac{1 + \omega_n\cos(\frac{2\pi}{n}(j-i)) + \omega_n\cos(\frac{2\pi}{n}(j-i+1))}{n}.$$

Since $n \geq 3$, $a_{i,j} \geq 0$ and $\sum_{j=1}^{n} a_{i,j} = 1$, $i = 1..n$ and one checks by substitution that

$$A_i = \sum_j a_{i,j}C'_j.$$

■

If it were not for the contraction by $\omega_n \in (\frac{1}{2}, 1]$, the map from the C'_i to the C_i would be a projector. Unfortunately, the contraction is necessary as the following, admittedly extreme example proves. If $C'_j := [-1,0,0]$ for $j = 1..n-1$ and $C'_n := [n-1,0,0]$, then $C_i = \frac{2\omega_n}{n}\cos(\frac{2\pi}{n}i)[n-1,0,0]$ and the constraint that A_i lie in the convex hull is $\frac{n-1}{n}\omega_n(-\cos(\frac{2\pi}{n}(i-1)) - \cos(\frac{2\pi}{n}i)) < 1$.

Since all coefficients are computed as convex averages of the C_i, the following proposition holds.

(4.2) Proposition. *If the construction uses three-sided patches at irregular vertices, i.e. Steps 4T, 5T, 6T or 6M, then the surface is in the convex hull of the input mesh.*

(4.3) Example: The following example shows, that Step 6R does not always yield

planar surfaces for planar cells and therefore does not stay within the convex hull of the input mesh. Consider the corner $(0,0,0)$ of a cube with neighbor vertices $V_1 = (1,0,0)$, $V_2 = (0,1,0)$, and $V_3 = (0,0,1)$. Then

$$P_{11,1} = E_2 - E_3 + E_1 = \frac{1}{9}\begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$$

is outside the cube. This problem can be fixed by using biquintic patches.

(4.4) Proposition. *The curvature at V is zero if and only if the neighboring vertices of the refined mesh, $V_i, i = 1..n$ lie in the same plane as V.*

Proof Let $P(n)$ be the component of P normal to the tangent plane at V. If $V_i(n) \neq 0$ for some i, then the curvature of the ith boundary curve is nonzero. Conversely, if all V_i lie in the tangent plane, then all boundary curves are coplanar. In the case of four-sided patches this implies $E_i(n) = 0$ and hence $P_{11,i}(n) = 0$. In the case of three-sided patches this implies $P_{111,i}(n) = \frac{1+c}{6}V_i(n) = 0$. ■

Zero cut ratios generate a singular parametrization at the input mesh point. Just as for B-splines with coalesced knots, the degree of smoothness drops by one. This has the desirable consequence that the C^1 surface degenerates into a C^0 surface that tightly interpolates the input mesh.

(4.5) Proposition. *An edge between two cells with zero transversal cut ratios is interpolated. Planar cells with zero cut ratios are covered by a planar surface if represented with three-sided patches.*

Proof Zero cut ratios coalesce all cell centers C_i surrounding a mesh point V into V. The boundary curves are convex averages of the C_i, hence coincide with the edges. Up to Step 5 (T,R or M), all coefficients are convex combinations of points on the edges and planarity follows. For a three-sided or mixed construction

$$Q_{020,i} = Q_{110,i} = Q_{011,i} = Q_{110,i+1} = V \qquad Q_{200,i} = Q_{110,i-1} = V_i$$
$$Q_{101,i} = \frac{Q_{200,i} + Q_{020,i}}{2} = \frac{V_i + V}{2} \qquad P_{111,i} = \frac{(5-c)V + (1+c)V_i}{6}$$

proving planarity. ■

To model with the vector space it is good to know that the resulting surfaces can interpolate certain averages of the input data.

(4.6) Proposition. *Let $a_{i,j}$, $i = 1..n$, $j = 1,2$ be the blend ratios and C_i the centers of the n subcells surrounding a cell centroid V. If $a_{i,j} = a_j$ and $C_i = C_i'$, then the surface generated by the algorithm interpolates the cell centroid.*

Proof Let V_i be the vertices of the cell. Step 3 determines the vertex of the refined mesh corresponding to the cell as

$$\begin{aligned}\bar{V} &= \frac{1}{n}\sum C_i \\ &= \frac{1}{n}\sum[(1-a_{i1})(1-a_{i2})V_i + (1-a_{i1})\frac{a_{i2}}{2} + (1-a_{i-1,1})\frac{a_{i-1,2}}{2} \\ &\quad + (1-a_{i2})\frac{a_{i1}}{2} + (1-a_{i+1,2})\frac{a_{i+1,1}}{2} + \frac{1}{n}\sum_l a_{l,1}a_{l,2}]V_i\end{aligned}$$

In the particular case, the bracketed expression sums to one. Since $\bar{V}$ is a vertex coefficient of the patches meeting at $\bar{V}$ it is interpolated. ■

Thus a uniform choice of blend ratios suffices to interpolate the centroids of a cube (cf. A.2 of the Appendix).

5. Conclusion

The algorithm defines a surface representation that generalizes quadratic C^1 splines. It smoothes a general, regular or irregular mesh of points into a C^1 surface with a quadratic-cubic parametrization. The user may choose either four-sided or a three-sided or a mixed patch representation. In the case of a purely four-sided representation some continuity and shape properties are traded for low degree and simplicity of construction. Where the mesh is regular, the surface is quadratic. Input meshes with the same connectivity and the same blend ratio for corresponding cells give rise to a vector space of smooth surfaces. That is, it suffices to add and subtract the control meshes in order to add and subtract the corresponding surfaces and the result is again a smooth surface. This and the fact that the convex hull property holds is useful for approximating and locally editing the spline surface. The role of the knot spacing is played by geometrically intuitive blend ratios. Zero blend ratios result in a C^0 surface that tightly interpolates the input mesh. It is possible to interpolate the input mesh without solving a global sparse system of equations. Here the analogy is interpolation by a quadratic spline at every second knot rather than the cubic Catmull-Rom spline construction. The construction can easily be extended to a rational representation by using a fourth coordinate to generate the coefficients of the denominator polynomial.

References

Boehm, W., G. Farin, J. Kahmann (1984), A survey of curve and surface methods in CAGD, *CAGD* **1**, p. 43.

C. W. de Boor, K. Höllig, S. Riemenschneider (1992), *Box splines*, Springer Verlag, NY, to appear.

E. Catmull, J. Clark (1978), Recursively generated B-spline surfaces on arbitrary topological meshes, *Computer Aided Design* **10**, No 6: 350–355.

W. Dahmen, C.A. Micchelli, H.P. Seidel (199x), Blossoming begets B-splines bases built better by B-patches, *Math. of Computation*, to appear.

D. Doo (1978), A subdivision algorithm for smoothing down irregularly shaped polyhedrons, *Proceedings on interactive techniques in computer aided design*, Bologna: 157–165.

N. Dyn, D. Levin, D. Liu (1992), Interpolatory convexity preserving subdivision schemes for curves and surfaces, preprint.

G. Farin (1990), Curves and surfaces for computer aided geometric design, Academic Press.

T.N.T. Goodman (1988), Closed surfaces defined from biquadratic splines, Constructive Approximation 7 1991, 149-160.

J.A. Gregory (1990), Smooth parametric surfaces and n-sided patches, *Computation of curves and Surfaces*, W. Dahmen, M. Gasca and C.A. Micchelli, eds., Kluwer Academic Publishers, Dordrecht, 1990: 457–498.

J. M. Hahn (1989), Filling polygonal holes with rectangular patches, *Theory and Practice of geometric modeling*, W. Straßer and H.-P. Seidel eds., Springer 1989.

K. Höllig, H. Mögerle (1989), G-splines, *Computer Aided Geometric Design* **7**: 197–207.

C. Loop (1987) Smooth subdivision surfaces based on triangles, Master's thesis, University of Utah.

C. Loop, T. DeRose (1990), Generalized B-spline surfaces of arbitrary topology, *Computer Graphics* **24**(4): 347–356.

C. T. Loop (1993), Smooth low degree polynomial spline surfaces over irregular meshes, preprint, Jan 1993.

J. Peters (1990), Smooth Mesh Interpolation with Cubic Patches, CAD, **22** No 2 (1990), 109-120.

J. Peters (1991), Smooth Interpolation of a Mesh of Curves, *Constructive Approximation*, **7** (1991), 221-247.

J. Peters (1992a), Constructing C^1 surfaces of arbitrary topology using biquadratic and bicubic splines, May 1992, to appear in *Designing fair curves and surfaces*, N. Sapidis (ed.).

J. Peters (1992b), Smooth free-form surfaces over irregular meshes generalizing quadratic splines, Sept 1992, to appear in CAGD.

M. Sabin (1983), Non-rectangular surface patches suitable for inclusion in a B-spline surface, in P. ten Hagen (ed.), *Proceedings of Eurographics '83*, North Holland, 57–69.

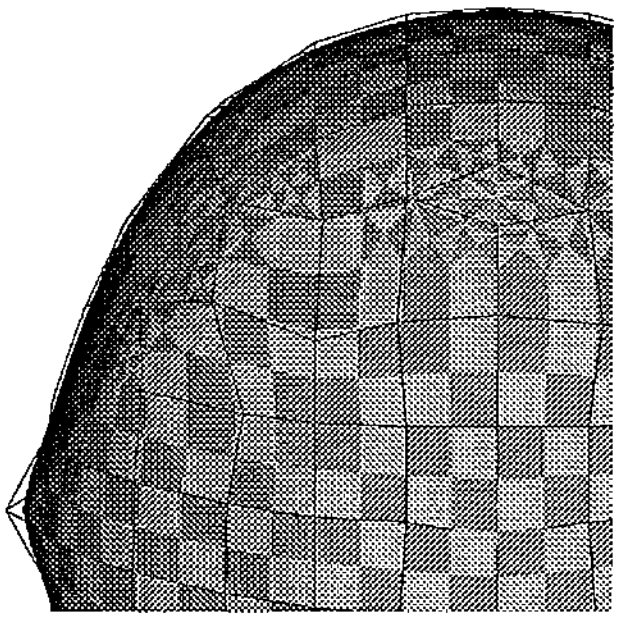
R.F. Sarraga (1987), G^1 Interpolation of Generally Unrestricted Cubic Bézier curves, *CAGD* 4(1-2):23–40,1987

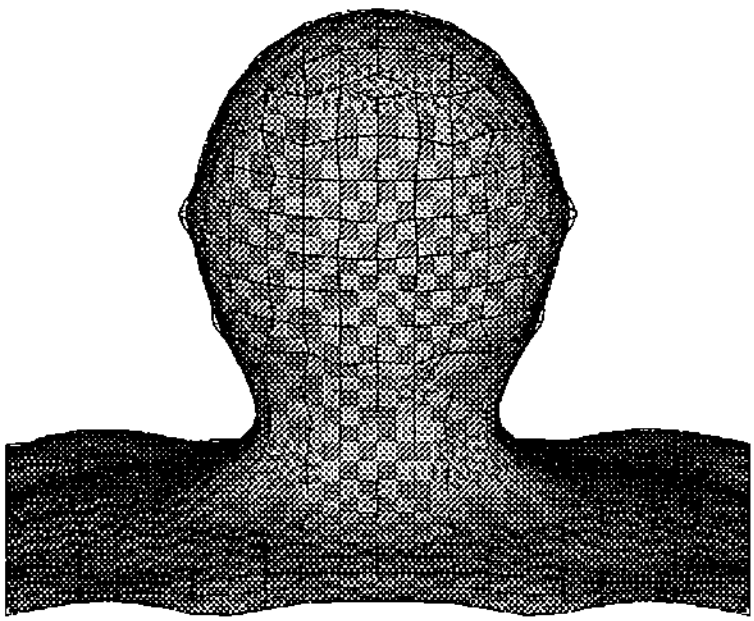
H.-P. Seidel (1991), Symmetric recursive algorithms for surfaces: B-patches and the de Boor algorithm for polynomials over triangles, *Constr. Approx.* **7**: 257–279.

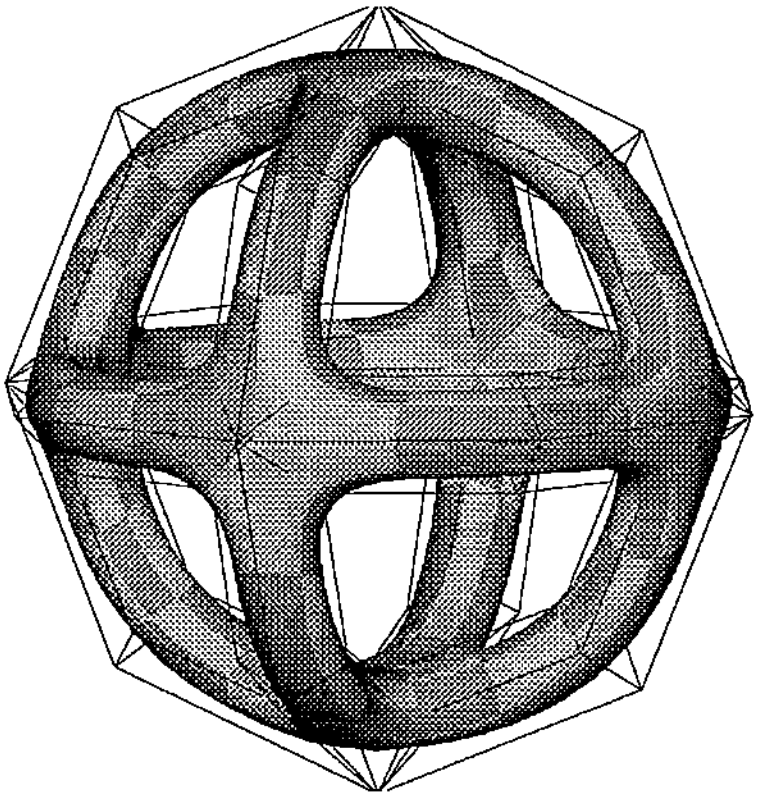
J.J. van Wijk (1984), Bicubic patches for approximating non-rectangular control-point meshes, *Computer Aided Geometric Design* **3**, No 1: 1–13.

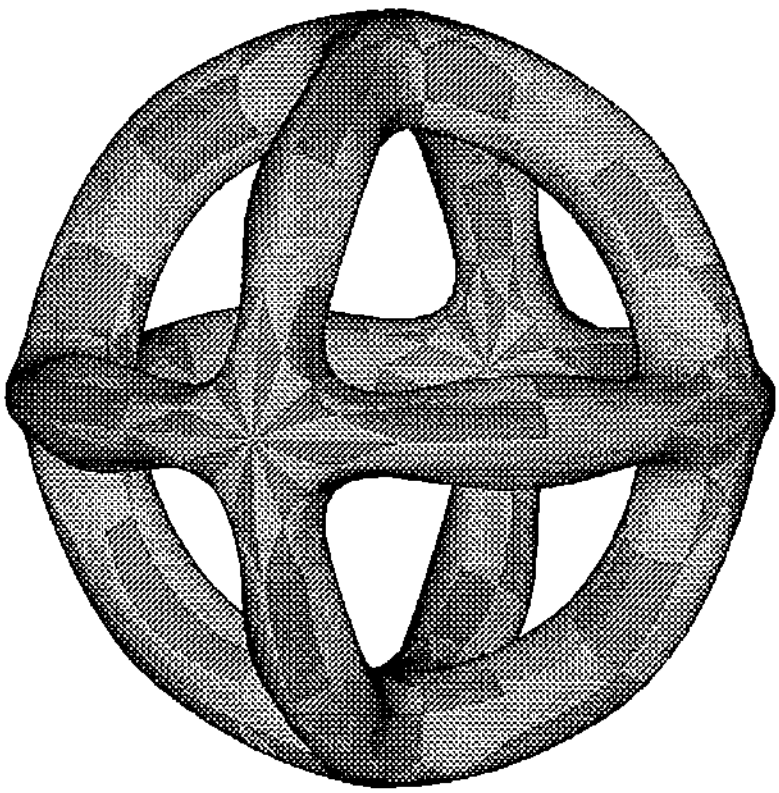
A. Examples

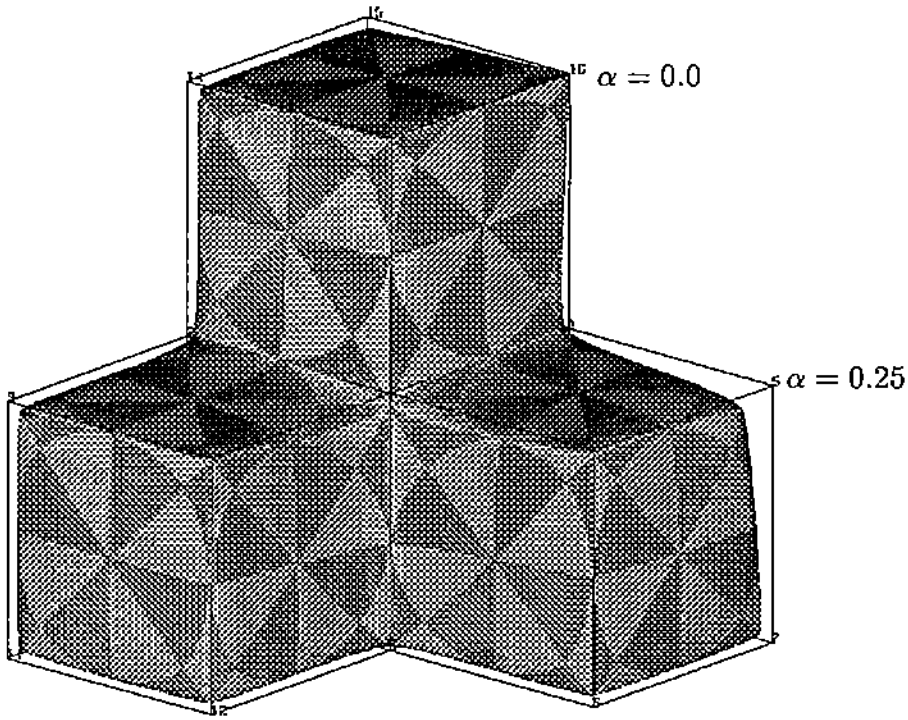
Three examples illustrate the flexibility of the algorithm with respect to the topological structure. the blend ratio and interpolation.











A1. The first two figures show a geological formation and a detail of the formation. The surface is constructed with a mix of three-sided and four-sided patches.

A2. The second two figures show a multi-handle object. The top surface uses only four-sided patches and does not interpolate the input mesh. The bottom surface is constructed using a mix of three-sided and four-sided patches. It interpolates the mesh points. One can also interpolate with a surface consisting only of four-sided patches.

A3. The last object, constructed from three-sided patches only, illustrates selective blending. The overall blend ratio is 0.1. However at the indicated mesh points the ratio is changed to 0.25 and 0.0 respectively. Ratios can also be selectively changed to blend edges with different radius.