Smooth free-form surfaces over irregular meshes generalizing quadratic splines

by

Jörg Peters[†]
jorg@cs.purdue.edu

Key words: C^1 surface, vertex enclosure, reparametrization, edge cutting, box splines, NURBS, blending

Running title: Smooth free-form surfaces over irregular meshes

Version: Feb 03 93 **Date printed:** February 4, 1993

Submitted to: Proceedings of the Oberwolfach conference June 1992, J. Hoschek (ed.)

Abstract

An algorithm for refining a combinatorially unrestricted mesh of points into a bivariate C^1 surface is given. The algorithm generalizes the construction of quadratic splines from a mesh of control points. When the mesh is regular then a quadratic spline surface is generated. In general, an explicit parametrization of the surface with quadratic and cubic pieces is given. Irregular input meshes with non quadrilateral mesh cells and more or fewer than four cells meeting at a point are allowed. Consequently, the algorithm can model bivariate open or closed surfaces of general topological structure.

1. Introduction

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 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 mesh

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

bounding 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 an algorithm that removes the regularity restrictions from the input mesh and yields a more 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.

- 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.
- 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.
- The surface is parametrized by low degree polynomial patches. The representation can be extended to rational patches.
- It is possible to interpolate the input mesh points and normals without solving a system of constraints.
- The coefficients of the parametrization can be obtained by applying averaging masks to the input mesh. Thus the algorithm can be interpreted as a rule for cutting an input polytope such that the limit polytope is the spline surface.
- 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.
- 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.

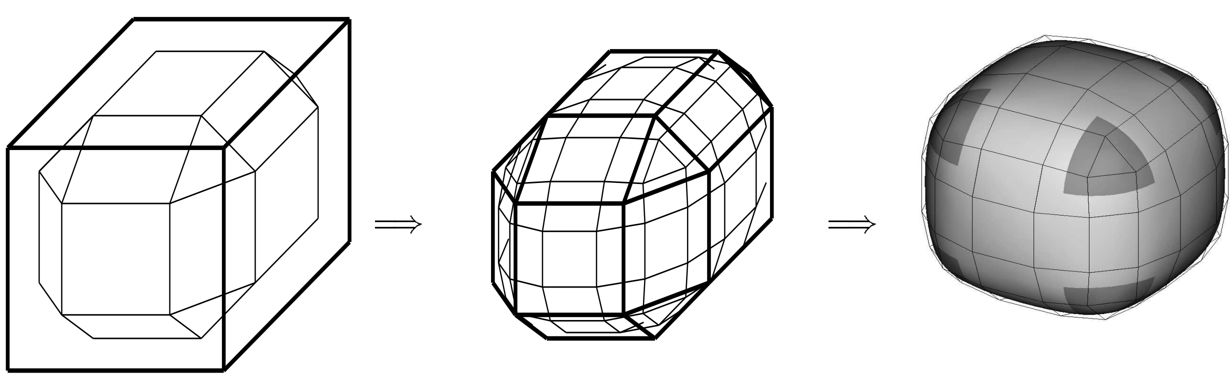


Figure 1.1: The cutting paradigm applied to a cube. The light regions of the output surface are covered by a quadratic spline, the dark regions by cubics.

The algorithm described in this paper generalizes the quadratic C^1 spline paradigm to generate surfaces with the above properties. The central idea is to refine the irregular input mesh by a simple, averaging process and generate strips of regular mesh points that isolate regions of irregular points. Using this approach, [Peters '92] generates surfaces that consist of strips of biquadratic tensor-product splines complemented by bicubic patches to cover the isolated irregular mesh regions. The construction detailed in this paper uses the same cutting paradigm to generate the control points of symmetric piecewise quadratic box splines over the four direction mesh[1] and fills the remaining holes with cubic triangular patches. Remarkably, filling in the cubics smoothly does not require solving systems of constraints but only averaging the box spline control points. When applied to a hole with four edges, quadratic patches equivalent to the box spline surface are generated. Thus the entire surface can be given a uniform representation in terms of triangular quadratic and cubic patches in Bernstein-Bézier form.

The algorithm is in part motivated by algorithms for generalized subdivision ([Sabin '76], [Doo '78], [Catmull and Clark '78], [Loop '87], [Dyn, Levin and Liu '92], etc.). The algorithm does a two stage generalized subdivision. The main difference between the algorithm and earlier schemes is that the earlier schemes 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 CAGD representations. A second motivation is the work on G-spline spaces by [Sabin '83], [Goodman '88] and [Höllig, Mögerle '89]. The present algorithm defines a G-spline space that is more localized and does not need large unstructured sparse systems of equations to match data. This makes it easier to reason about the shape of the resulting surface. A third foundation of the algorithm is the work on reparametrization and geometric smoothness (see e.g. [Gregory '90] for a survey). The algorithm of this paper differs from schemes like [Sarraga '87],[Hahn '89] in that no constraint system needs to be solved to enforce patch to patch smoothness. The vertex enclosure problem of joining surface pieces at a common point [Peters '91] is solved in a simple and natural fashion. While the present algorithm generates only tangent plane continuous surfaces, B-patches ([Seidel '91], [Dahmen, Micchelli, Seidel '9x]) offer kth order continuity for patches of degree $k+1$. However, currently B-patches are slow to evaluate, suffer from an irregular choice of knots that influences the shape of the surface, and are restricted in their modeling of closed surfaces. In comparison, the surfaces generated by the algorithm of this paper can be evaluated by subdivision, have intuitive cut ratios in lieu of knot spacings and can model arbitrary free-form objects. S-patches, as presented in [Loop, DeRose '90] have a non standard representation and slow evaluation of the rational surface pieces when the number of edges is large. Similar arguments apply to Gregory patches [Gregory '74].

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

[1] Both the Bernstein-Bézier form and box splines are standard tools of geometric modeling. See for example [Boehm, Farin, Kahmann '84] and [de Boor, Höllig, Riemenschneider 92]. The quadratic box spline can be traced back to [Zwart '73] and [Powell '74].

and Appendix gives three examples.

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

The three steps of the algorithm are:

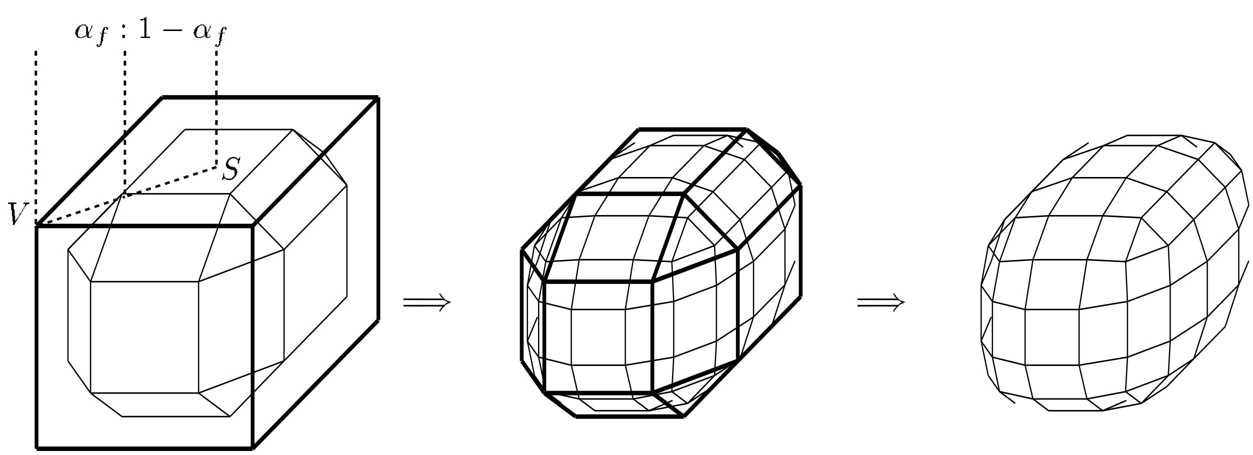
- Refining the input mesh to generate the control points of the box spline surface.
- (Optional) Converting the box spline surface into Bernstein-Bézier form to make the representation uniform.
- Covering the remaining isolated holes with cubic patches.

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 edges to a cell or the number of cells meeting at a vertex. The mesh may model a bivariate open or closed surface of arbitrary topological structure. Each cell f of the input mesh has a shape parameter α_f, called *blend ratio*. The blend ratio is a number between zero and one. A smaller ratio results in a surface that follows the input mesh more closely and changes the normal direction more rapidly close to the mesh edges. The *output* of the algorithm are the Bernstein-Bézier coefficients of quadratic and cubic patches that parametrize a tangent-plane continuous surface. The surface interpolates the centroid, the average of the vertices, of each cell of the input mesh.

1. Refining the input mesh to generate the control points of the box spline surface. Apply two steps of Doo-Sabin's averaging procedure [Doo '78]: at each step, s new points are created for each s-sided cell. Each new point connects to two new points generated at the two adjacent vertices of the same cell and the two adjacent cells of the same old mesh point. A new point corresponding to a vertex V of the cell f with centroid S has the coordinates

$$(1-\alpha_f)V + \alpha_f S$$

By default, the ratio of a cell in the second step is the average of the ratios of the old cells that contribute a vertex in the first step.



After refining the mesh, each mesh point is surrounded by four cells. If all four cells have exactly four edges, then the nine mesh points defining the cells can be interpreted as the control mesh of a box spline.

2. (Optional) Converting the box spline surface into Bernstein-Bézier form. This step may be omitted. It serves only to unify the surface representation in Bernstein-Bézier form. By symmetry it suffices to give the conversion formulae for the following four Bernstein-Bézier coefficients in terms of the box spline control points.

$$
\begin{aligned}
Q_{002} &= \frac{1}{8}(C_1 + C_2 + C_3 + C_4 + 4A) \\
Q_{011} &= \frac{1}{8}(2C_2 + 2C_3 + 4A) \\
Q_{020} &= \frac{1}{8}(2C_2 + 2C_3 + 2A + 2B_2) \\
Q_{110} &= \frac{1}{8}(4C_2 + 4A)
\end{aligned}
$$

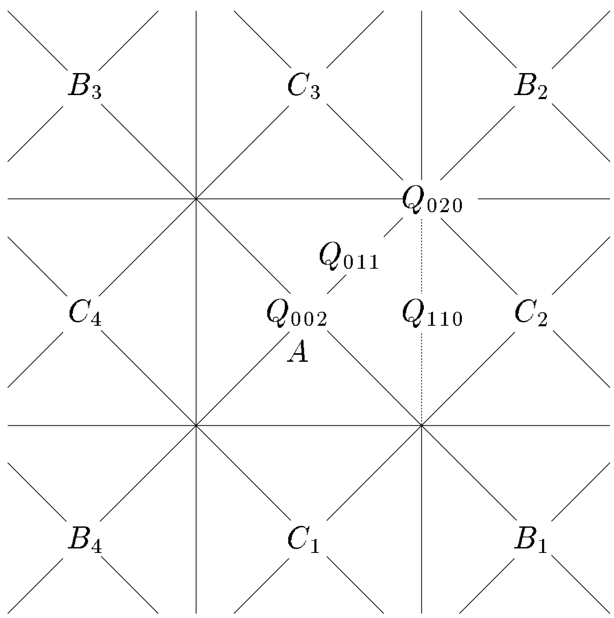


Figure 2.1: Converting the box spline control mesh with points A, B_i, C_i into a mesh of Bézier coefficients Q_{ijk}.

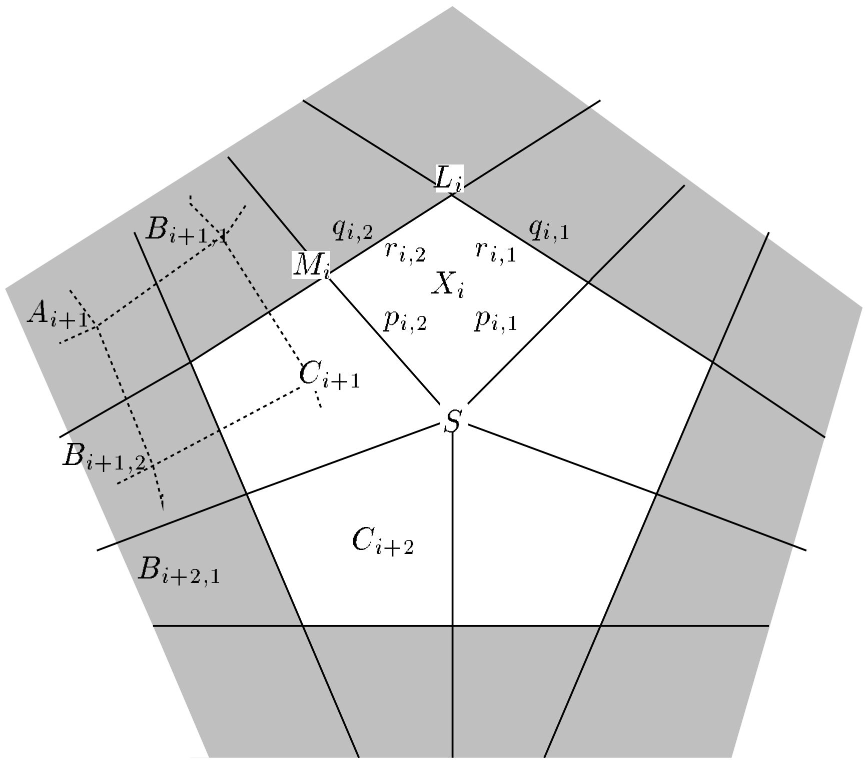


Figure 2.2: Each quadrilateral of the 5-sided cell is covered by 4 triangular patches: $r_{i,1}$, $p_{i,1}$, $p_{i,2}$, $r_{i,2}$. – A_i, B_{ij}, C_i are box spline control points as in Figure 2.1.

3. Covering the remaining isolated holes with cubic patches. Each non quadrilateral, s-sided mesh cell is divided into quadrilaterals $i = 1..s$ each of which is covered by four triangular patches, p_{ij} and r_{ij}, $j = 1, 2$, with Bernstein-Bézier coefficients $P_{klm,ij}$, and $R_{klm,ij}$. If Step 2 is executed, then the cell is surrounded by $2s$ patches $q_{i,j}$ with coefficients $Q_{klm,ij}$ as shown in Figure 2.3. First one determines S, the centroid of the mesh cell and various auxiliary vectors:

$$S := \frac{1}{s}\sum_{i=1}^{s} C_i, \qquad E_i := \frac{C_i + C_{i+1}}{2}, \qquad M_i := \frac{C_i + C_{i+1} + B_{i+1,1} + B_{i,2}}{4},$$

$$F_{i,j} := \frac{C_i + B_{i,j}}{2}, \qquad L_i := \frac{C_i + B_{i,1} + B_{i,2} + A_i}{4}.$$

The boundary curves of the cell are degree raised quadratics:

$$R_{030,i2} = L_i, \quad R_{120,i2} = \frac{1}{3}(L_i + 2F_{i,2}), \quad R_{210,i2} = \frac{1}{3}(M_i + 2F_{i,2}), \quad R_{300,i2} = M_i.$$

The tangent coefficient P_{210} is chosen by extending the box spline.

$$P_{210,i2} = \frac{2}{3}E_i + \frac{1}{3}M_i.$$

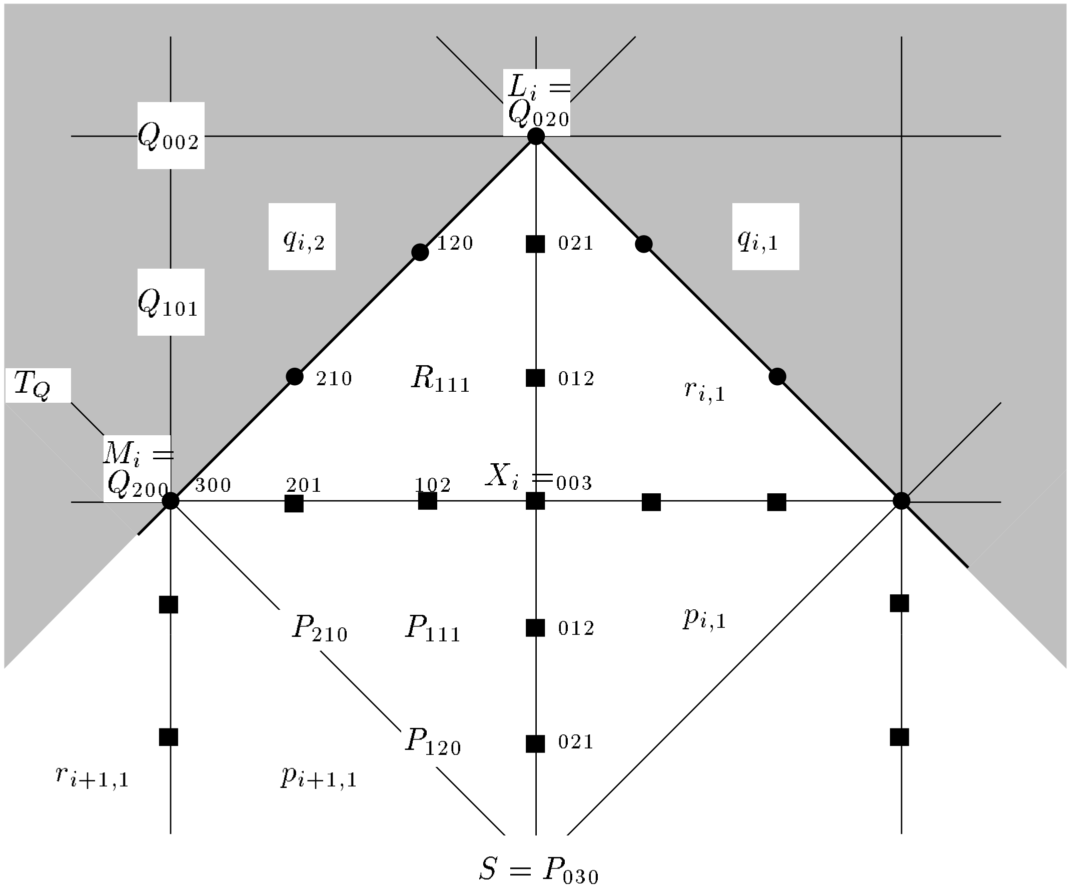


Figure 2.3: Enlargement of the corner L_i.
The cubic construction consists mainly of degree-raising (•) and averaging (■).
$R_{111} = R_{111,i,2}$, $P_{111} = P_{111,i,2}$, $P_{210} = P_{210,i,2} = P_{210,i+1,1}$, $P_{120} = P_{120,i,2} = P_{120,i+1,1}$.

The tangents of the curves M_iX_i and L_iX_i are computed as averages weighed by $c := \cos(\frac{\pi}{s})$:

$$R_{201,ij} = c^2 R_{210,ij} + (1-c^2) P_{210,ij}, \quad R_{021,ij} = \frac{1}{2} R_{120,i2} + \frac{1}{2} R_{120,i1}, \text{ and}$$

$$R_{111,i2} = R_{120,i2} + \frac{1}{12}(C_{i+1} - B_{i,2} + 2(1-c^2)(C_i - A_i)).$$

The tangent coefficients $P_{120,i}$ at S are generated by applying a discrete first order Fourier filter as in [Loop '90, Fig.3], [Van Wijk '86 (14)]:

$$P_{120,ij} = S + \frac{4\alpha}{3s} \sum_{l=1}^{s} \cos(\frac{2\pi}{s} l) E_{i+l}.$$

Here $\alpha \in (0,1)$ is the shape parameter of the cell analogous to the blend ratio in the mesh refinement; $\alpha = 1/(1+c)$ is the recommended value. Finally,

$$R_{021,i2} = R_{021,i1} = \frac{1}{2}R_{111,i2} + \frac{1}{2}R_{111,i1},$$
$$P_{021,i2} = P_{021,i1} = \frac{1}{2}P_{120,i2} + \frac{1}{2}P_{120,i1},$$
$$P_{111,i2} = (1-c^2)P_{120,i2} + c^2 P_{210,i2} + \frac{1}{4}(P_{201,i2} - P_{201,i+1,1} + P_{021,i2} - P_{021,i+1,1}),$$
$$P_{012,i2} = P_{012,i1} = \frac{1}{2}P_{111,i2} + \frac{1}{2}P_{111,i1},$$
$$P_{102,i2} = R_{102,i2} = (1-c^2)P_{111,i2} + c^2 R_{111,i2},$$
$$P_{003,i2} = P_{003,i1} = R_{003,i2} = R_{003,i1} = \frac{1}{2}P_{102,i2} + \frac{1}{2}P_{102,i1}.$$

By symmetry, this determines all the coefficients of the cubic patches, e.g. $P_{111,i1} = (1 - c^2)P_{120,i1} + c^2 P_{210,i1} + \frac{1}{4}(P_{201,i1} - P_{201,i-1,2} + P_{021,i1} - P_{021,i-1,2})$.

Extensions of the Algorithm:

- A different combination of blend ratios can be used in each step and the blend ratios may be individually changed and associated with edges rather than with cells. The above choice is just a default to simplify the exposition.
- A vertex V of the input mesh can be interpolated as follows. After the first of the two refinement steps, move the points P_i, $i = 1..s$ of the refined mesh constructed around V by $V - \frac{1}{s}\sum_{i=1}^{s} P_i$. Then V is the centroid of the resulting cell and will be interpolated by Proposition 3.5. Similarly, one can interpolate normals at the vertices.
- To obtain rational surfaces, one simply associates a fourth coordinate with each box spline control point and applies the algorithm to points in $\mathbb{R}^4$. The fourth coordinate corresponds to the coefficient of the rational weight function (cf. Corollary 3.3).
- Like a NURB surface, an open surface generated by the algorithm has a piecewise quadratic rim. A piecewise curve that closely follows the outlines of the original input mesh rim can be interpolated with the following ad hoc measure. Extend the boundary patches by reflection, and replace the outermost layer of coefficients of the extrapolated patches with the coefficients of the piecewise curve. Then average the coefficients second layer of coefficients from the rim to smooth out the result.

3. Continuity and vector space properties

This section shows that splines generated by the surface form a smooth vector space for fixed blend ratios and connectivity. We define smoothness as 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 to avoid cusps. We prepare the result with the following lemma.

(3.1) Lemma. *The choice of the coefficients P_{111} minimizes the deviation from the center coefficient of a degree-raised quadratic subject to the continuity constraints.*

Proof Define P^*_{111} as the central coefficient of a degree-raised quadratic based on given boundary data in Step 3 of the algorithm:

$$P^*_{111} := \frac{1}{4}(2P_{201} + 2P_{021} + P_{120} + P_{210} - P_{300} - P_{030}).$$

We want to minimizes $(P_{111,i,1} - P^*_{111,i,1})^2 + (P_{111,i,2} - P^*_{111,i,2})^2$ subject to the C^1 constraint $\frac{1}{2}P_{111,i+1,1} + \frac{1}{2}P_{111,i,2} = (1-c^2)P_{120,i,2} + c^2 P_{210,i,2}$. By Lagrangian duality, this minimization problem is solved by

$$P_{111,i,2} = (1-c^2)P_{120,i,2} + c^2 P_{210,i,2} + \frac{1}{2}(P^*_{111,i,2} - P^*_{111,i+1,1})$$

as claimed. ■

(3.2) Theorem. *The surface constructed by the algorithm in Section 2 is C^1.*

Proof With $c := \cos(\frac{\pi}{s})$. and $a := \frac{c^2}{1-c^2}$, the transitions between the patches are as follows (cf. Figure 2.2).
(1) Across the edges L_iM_i, $r_{i,j}$ is constructed by enforcing $r_{i,j}(E_j) = q_{i,j}(\phi_{i,j}(E_j))$ and $D_j r_{i,j} = D_j(q_{i,j} \circ \phi_{i,j})$, where t_1 parametrizes the edge L_iM_i, t_2 the edge L_iM_{i-1} and

$$\phi_{i,1} := \text{id} + t_1 t_2 \begin{bmatrix} 1-2c^2 \\ 0 \end{bmatrix}, \quad \phi_{i,2} := \text{id} + t_1 t_2 \begin{bmatrix} 0 \\ 1-2c^2 \end{bmatrix}.$$

(2) Across the edges M_iS, $D_1 p_{i,2} = D_1(p_{i+1,1} \circ \psi_i)$, where t_1 parametrizes the edge M_iL_{i+1}, t_2 the edge M_iS and

$$\psi_i := \text{id} + t_1 \begin{bmatrix} 0 \\ 1-2c^2 \end{bmatrix}.$$

(3) Across the edges L_iX_i and X_iS, the reparametrization is the identity and (4) across the edges M_iX_i and $M_{i-1}X_i$, it is the scaling $\text{id}^a := a * \text{id}$. Due to the simplicity of the maps

(3) and (4), we may concentrate on the construction of P_{210}, R_{111}, P_{111} and P_{120}. Writing $[b_0,\ldots,b_d]$ for $p : t \mapsto \sum_{j=0}^{d} t^j(1-t)^{d-j}b_j$, the C^1 constraints $D_2 r_{i,2} = D_2(q_{i,2}\circ\phi_{i,2})$, across L_iM_i in Bernstein-Bézier form are

$$2[1,\ell][Q_{110}-Q_{020},Q_{200}-Q_{110}] =$$

$$3[R_{021}-R_{030},2(R_{111}-R_{120}),R_{201}-R_{210}]+2[1,m][Q_{011}-Q_{020},Q_{101}-Q_{110}],$$

where $Q_{200,i2}=M_i$, $Q_{101,i2}=(2B_{i,2}+C_i+B_{i+1,1})/4$, $Q_{011,i2}=(2B_{i,2}+C_i+A_i)/4$, and $Q_{110,i2}=F_{i2}$. Since $Q_{101}-Q_{110}=\frac{T_Q-Q_{110}}{2}$, $T_Q := (B_{i,2}+B_{i+1,1})/2$ and $R_{201}-R_{210} = \frac{2}{3}(P_{210}-R_{210})(1-c^2) = \frac{2}{3}(-T_Q-Q_{110})(1-c^2)$ and since T_Q-M and $Q_{110}-M$ are in general linearly independent, the third equation,

$$2\ell(M-Q_{110}-M) = m(T_Q-Q_{110})+2(-T_Q-Q_{110})(1-c^2),$$

implies $\ell = m = 2(1-c^2)$. Therefore the second constraint is

$$R_{111} = R_{120}+\frac{1}{3}(Q_{200}-Q_{110}-(Q_{101}-Q_{110})+\ell(Q_{110}-Q_{020}-(Q_{011}-Q_{020}))),$$

enforced by the construction.

Setting $S=0$, the coplanarity of the tangent coefficients P_{120} at S follows from

$$\begin{aligned}&\sum_{i=1}^{s}\cos(\frac{2\pi}{s}i)(P_{120,i-1}-2cP_{120,i}+P_{120,i+1})\\&=\sum_{i=1}^{s}P_{120,i}\big(\cos(\frac{2\pi}{s}(i-1))+\cos(\frac{2\pi}{s}(i+1))-2\cos(\frac{2\pi}{s})\cos(\frac{2\pi}{s}i)\big)=0.\end{aligned}$$

The choice of P_{111} in Lemma 3.1 enforces the remaining constraint $D_1p_{i,2} = D_1(p_{i+1,1}\circ\psi_i)$ across the edges M_iS:

$$\begin{aligned}2(1-c^2)&[P_{210,i,2}-P_{300,i,2},2(P_{120,i,2}-P_{210,i,2}),P_{030,i,2}-P_{120,i,2}]\\&=[P_{201,i,2}-P_{300,i,2},2(P_{120,i,2}-P_{210,i,2}),P_{021,i,2}-P_{120,i,2}]\\&+[P_{201,i+1,1}-P_{300,i+1,1},2(P_{120,i+1,1}-P_{210,i+1,1}),P_{021,i+1,1}-P_{120,i+1,1}].\end{aligned}$$

■

(3.3) Corollary. *The rational surface generated by applying the algorithm to vectors with four components and using the last component as the coefficient of the denominator is tangent plane continuous when the denominator does not vanish.*

Proof If $P,Q : [0..1]^2 \mapsto \mathbb{R}^3$ and $p,q : [0..1]^2 \mapsto \mathbb{R}$ and $P=Q$ and $p=q\neq 0$ along a boundary shared by the functions P/p and Q/q, then

$$D_1(\frac{P}{p}) = D_1(\frac{Q}{q}\circ\phi) \iff q(D_1P-D_1(Q\circ\phi)) = Q(D_1p-D_1(q\circ\phi))$$

along that boundary. When the algorithm is applied to the coefficients of $(P, p) \in \mathbb{R}^4$, then the second factor on either side of the equation vanishes and hence the transition is tangent plane continuous. ■

An important property of splines is that it suffices to add and subtract the control meshes in order to add and subtract the corresponding surfaces provided the knot spacings agree. For the spline surfaces generated by the present algorithm the role of the knot spacings is played by the blend ratios.

(3.4) Theorem. *C^1 surfaces generated from input meshes with the same connectivity and the same blend ratio for corresponding cells form a vector space.*

Proof If $P = \sum \alpha_i P_i$ and $Q = \sum \alpha_i Q_i$ are are points of the refined mesh constructed with the same blend ratios α_i, then $P + Q = \sum \alpha_i (P_i + Q_i)$ as the additive vector space property requires. Two surfaces generated from input meshes with the same connectivity have a natural 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. ■

To model with the vector space it is good to know that the resulting surfaces interpolate averages of the input mesh points.

(3.5) Proposition. *The surface generated by the algorithm interpolates the centroids of the cells of the input mesh.*

Proof In the regular control mesh regions, the coefficients Q_{020} generated in Step 2 of the algorithm are both the centroid of a quadrilateral mesh cell and a vertex Bernstein-Bézier coefficient. In the regular control mesh regions, Step 3 explicitly chooses the centroid to be the vertex Bernstein-Bézier coefficient S. ■

4. Shape properties of the resulting surface

Symmetry and regularity of the input mesh lead to a a simpler parametrization. This point is made by the first two propositions, while the third shows that the edges of the input mesh are interpolated and thus the outlines of the input polytope recaptured when the blend ratios are zero.

(4.1) Proposition. *If the control points C_i of a mesh cell form an affine s-gon and $\alpha = 1$, then the boundary curves M_iS are quadratic.*

Proof Let L be an affine map such that $L^{-1}E_i = \begin{bmatrix} \cos(\frac{2\pi}{s}i) \\ \sin(\frac{2\pi}{s}i) \\ 0 \end{bmatrix}$. Then $P_{210} = M_i + \frac{2}{3}(E_i - M_i)$ and $P_{120} = L(0 + \frac{4}{3s}\sum\cos(\frac{2\pi}{s}j)L^{-1}E_i) = S + \frac{2}{3}E_i$ are the coefficients of the degree raised quadratic with coefficients M_i, E_i and $S = 0$. ■

(4.2) Proposition. *If Step 3 of the algorithm is applied to a 4-sided mesh hole, then the patches generated are quadratic.*

Proof If $s = 4$, then $\phi_{i,j} = \psi_i = \mathrm{id}^a = \mathrm{id}$. Since $P_{120,i} = S + \frac{2}{3}(\frac{C_i + C_{i+1}}{2} - S)$, the boundary curve M_iS is a quadratic with coefficients M_i, E_i and S. Rewriting the expressions in terms of the control points A_i, $B_{i,1}$, $B_{i,2}$ and C_i shows that

$$\frac{E_i + Q_{110,i,2}}{2} = R^*_{101} := Q_{200} + (Q_{110} - Q_{101})$$
$$\frac{Q_{110,i,2} + Q_{110,i,1}}{2} = R^*_{011} := Q_{020} + (Q_{110} - Q_{011})$$

Set $R^*_{110} := Q_{110}$, then

$$R_{111} = \frac{1}{3}(Q_{110} + Q_{200} + (Q_{110} - Q_{101}) + Q_{020} + (Q_{110} - Q_{011}))$$

is the center coefficient of a degree-raised quadratic patch r^*. Similarly one checks that

$$P_{111,i,2} = \frac{1}{2}P_{120,i,2} + \frac{1}{2}P_{210,i,2} + \frac{1}{4}(P_{201,i,2} - P_{201,i+1,1} + P_{021,i,2} - P_{021,i+1,1})$$

is the center coefficient of a degree-raised quadratic patch p^* with boundary coefficients $P^*_{110} := E_i$, $P^*_{011} := \frac{E_i + E_{i-1}}{2}$ and $P^*_{101} := R^*_{101}$. Since the remaining coefficients corresponding to the curves emanating from X_i are computed by averaging, the result follows. ■

One checks that, in the notation of Figure 2.1, the four triangular patches around A are a single quadratic if and only if $B_i - (C_i + C_{i+1}) = B_{i+2} - (C_{i+2} + C_{i+3})$ and that the surface is locally planar if two consecutive layers of coefficients lie in the same plane.

Zero cut ratios result is a singular parametrization at the input mesh point analogous to singularities in B-spline based curves or surface with coalesced knots. This has the desirable consequence that the C^1 surface degenerates into a C^0 surface that tightly interpolates the input mesh.

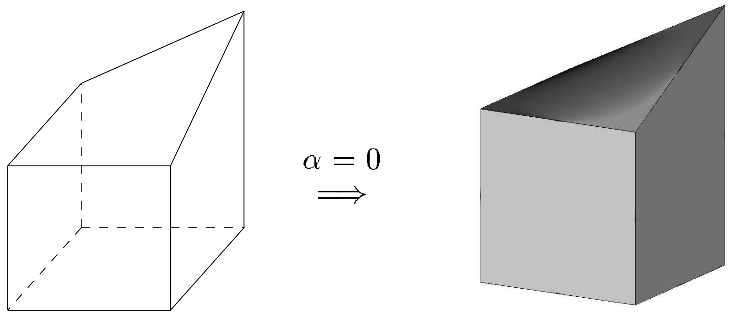


Figure 4.3: Zero blend ratios induce a C^0 surface that tightly interpolates the input mesh: a cube with the $(1,1,1)$-vertex replaced by $(1,1,2)$.

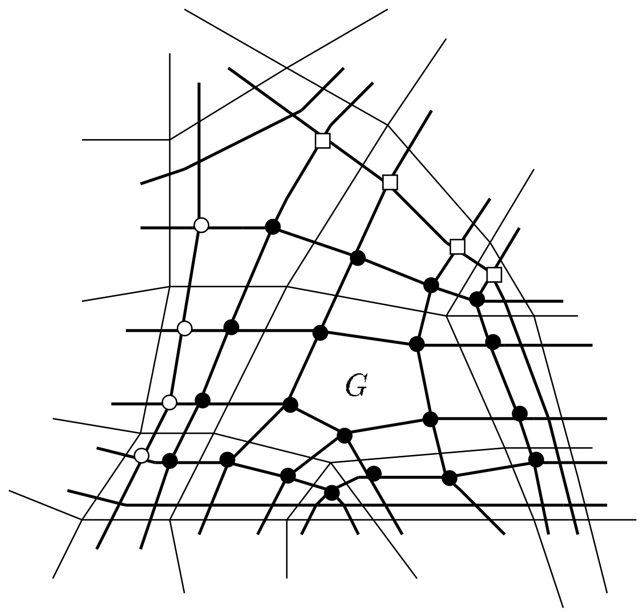


Figure 4.4: Zero blend ratios coalesce all control points • into the input mesh point G.

(4.5) Proposition. *An edge between two cells with zero cut ratios is interpolated. Planar cells with zero cut ratios are covered by a planar surface.*

Proof There can be two types of irregularities in the input mesh . A vertex irregularity originates from a vertex of the input mesh that does not have four neighbors. At a vertex irregularity, two layers of control points, namely A_i, B_{ij}, C_i for all i and j (cf. Figure 2.2) and hence the coefficients of the cubic patches are coalesced into A cell irregularity originates from a cell of the input mesh that is not quadrilateral. a single input mesh point. At a cell irregularity, the control points A_i, B_{ij}, C_i are coalesced into their respective cell vertex of the input mesh, say L_i. Considering Figures 2.1, and 2.2 this implies $B_3 = C_3 = B_2 = L_i$, and $C_4 = A = C_2 = B_4 = C_1 = B_1 = L_{i+1}$. Hence $Q_{002} = (7L_{i+1} + L_i)/8$, $Q_{011} = (3L_{i+1} + L_i)/4$, $Q_{020} = (L_{i+1} + L_i)/2$, $Q_{110} = L_{i+1}$; that is, the quadratic patches degenerate into straight lines $L_i\ M_i$ and $L_i\ M_{i-1}$ where $M_i = (L_{i+1} + L_i)/2$. If the irregularity corresponds to a planar cell, then the coefficients of the patches $r_{i,j}$ and $p_{i,j}$ are averages of vectors in that plane establishing the second claim. ■

5. Conclusion

The algorithm generalizes quadratic C^1 splines by removing the regularity restrictions of the control mesh. It smoothes a general mesh of points into a C^1 surface with a quadratic and cubic parametrization such that where the mesh is regular, the surface is quadratic. The role of the knot spacing is played by geometrically intuitive blend ratios. Input meshes with the same connectivity and the same blend ratio for corresponding cells give rise to a vector space of smooth surfaces. This is useful for locally editing the spline surface. Zero blend ratios result in a C^0 surface that tightly interpolates the input mesh. This is useful for locally controlled blending. Since interpolation is built into the spline space and since the construction is local, it is not necessary to solve a global sparse system of equations to match geometric data. The optional conversion of the box splines into Bernstein-Bézier form is an evaluation step that creates a large number of patches whose control mesh is closer to the surface than the box spline control mesh.

The key to extending the construction to smoother spline surfaces is the parametrization in the neighborhood of the irregular mesh points. Ideally such a parametrization is of low degree, depends only on the number of edges of the cell and becomes a spline if the number of edges agrees with the cells of the regular mesh. Methods that perturb the mesh of points or have no explicit but only a least squares solution are less desirable since it is difficult to predict the resulting shape.

Addendum: Charles T. Loop has independently been developing a similar quadratic-cubic algorithm [Loop '92].

References

W. Boehm (1983), Generating the Bézier points of triangular splines, R.E. Barnhill, W. Boehm (eds.), North Holland, 77–91.

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, *Report AA 886 University of Dundee.*

J.A. Gregory (1974), Smooth Interpolation without Twist Constraints, *Computer Aided Geometric Design*, 71-88, R.E. Barnhill, R.F. Riesenfeld, eds. , Academic Press.

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, 81–91.

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. Loop (1990) A G^1 triangular spline surface of arbitrary topology, manuscript, Dept. of CS and Eng., University of Washington, April 1990.

C. T. Loop (1992), private communication, September 17 1992.

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

J. Peters (1992), 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.).

M.J.D. Powell (1974), Piecewise quadratic surface fitting for contour plotting, in *Software for Numerical Mathematics*, D.J. Evans (ed.), Academic Press, London, 253–271.

M. Sabin (1976), The use of piecewise forms for the numerical representation of shape, PhD thesis, Hungarian Academy of Sciences, Budapest, Hungary.

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.

P.B. Zwart (1973), Multivariate splines with non-degenerate partitions, SIAM J. Numer. Anal. **10**, 665–673.