Smoothing Polyhedra Made Easy

JÖRG PETERS
Purdue University

A mesh of points outlining a surface is polyhedral if all cells are either quadrilateral or planar. A mesh is vertex-degree bounded if at most four cells meet at every vertex. This paper shows that if a mesh has both properties then simple averaging of its points yields the Bernstein–Bézier coefficients of a smooth, at most cubic, surface that consists of twice as many three-sided polynomial pieces as there are interior edges in the mesh. Meshes with checkerboard structure, that is, rectilinear meshes, are a special case and result in a quadratic surface.

Since any bivariate mesh and, in particular, any wireframe of a polyhedron can be refined, by averaging, to a vertex-degree-bounded polyhedral mesh the above allows reinterpretation of a number of algorithms that construct smooth surfaces and advertises the corresponding averaging formulas as a model for a wider class of algorithms.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*curve, surface, solid, and object representations*; J.6 [**Computer Applications**]: Computer-Aided Engineering—*computer-aided design*

General Terms: Algorithms

Additional Key Words and Phrases: Blending, box splines, C^1 spline mesh, C^1 surface, corner cutting, geometric continuity, polyhedral, vertex degree

1. INTRODUCTION

There are a variety of methods to give a smooth appearance to an object. For some animation purposes, it suffices to hide the seams between components or to average the normals of the surface polyhedra during the shading process. For more sophisticated use, a good strategy is to refine the object adaptively, by averaging the vertices of its boundary representation, as in the subdivisions algorithms of Doo [1978] and of Catmull and Clark [1978]. The purpose of this paper is to show that with comparable effort one can obtain the explicit polynomial representation of a smooth, namely, tangent continuous, surface that follows the outlines of a mesh of points such as the vertices and edges of a polyhedron. In particular, Section 2 gives a simple recipe for

This work was supported by NSF grant CCR-9211322.
Author's address: Department of Computer Science, Purdue University, West Lafayette IN 47907-1398.

smoothing vertex-degree bounded, polyhedral meshes whose definition is as follows:

—A mesh is *polyhedral* if all nonquadrilateral mesh cells are planar.

—A mesh is *vertex-degree bounded* if at most four cells meet at every vertex.

Smoothing these special meshes is of interest, because any surface mesh can be turned into a vertex-degree bounded, polyhedral mesh by refinement and projection. A typical example of a refinement that results in a vertex-degree bounded mesh is the application of a single step of the Doo–Sabin algorithm [Doo 1978]. Here, a new mesh point is created for each vertex of each cell; then each new point is connected to the two new points corresponding to the neighboring vertices of the same cell, and to the two new points corresponding to the neighboring cells and the same vertex. Consequently, each interior vertex has four neighbors at the end of the step. An example of a projection is the map $P_j \mapsto \sum_i P_i/n + 2\sum_i \cos(2\pi(i+j)/n)P_i/n$, for $j = 1 \dots n$ (see for example Loop [1987]), which maps n points $P_1, \dots, P_n$ into a common plane. Both operations can be scaled such that the basic shape of the mesh, and hence of the resulting surface, is preserved [Peters 1995].

A number of recent algorithms for geometric design [Lee and Majid 1991; Loop 1994; Peters 1992, 1993, 1995; Piah 1991] are directly or indirectly based on reduction to specific vertex-degree-bounded, polyhedral meshes (see, e.g., Figures 5a and 5b). Once such a mesh is obtained, the formulas of Section 2 apply (the result is shown in Figure 5c). Besides reinterpreting existing algorithms, the formulas can serve as a guide to creating meshes that can be efficiently smoothed or locally modified to be efficiently smoothable. For example, chopping off a high-degree vertex that is convex with regard to its neighbors bounds the vertex degree and creates locally a polyhedral mesh by a purely local operation. There are also meshes to which the smoothing algorithm can be applied directly. Buckminsterfullerene geodesic structures found in nature [Smalley 1991] are naturally vertex-degree-bounded polyhedral due to the valence and energy distribution of carbon-based structures.

This paper consists of the algorithm, its analysis, and examples. The analysis is made easy by the fact that the planar cells serve as tangent planes at user-specified points. Should the mesh additionally be regular, then the C^1 surface becomes the quadratic defined by these tangent planes, a four-direction box spline surface [de Boor et al. 1994]. For comparison, we note that the well-known algorithms of Piper [1987] and of Shirman and Sequin [1987] attack the more difficult problem of interpolating a mesh of curves. The polynomial degree of the resulting surfaces is therefore generally higher for these algorithms, and the surfaces are more likely to oscillate [Mann 1992, Sect. 2].

2. AN ALGORITHM FOR SMOOTHING A VERTEX-DEGREE-BOUNDED POLYHEDRON

The input to the algorithm are points W_l and their neighbor relationships. The points W_l form a mesh consisting of quadrilateral and planar cells, such

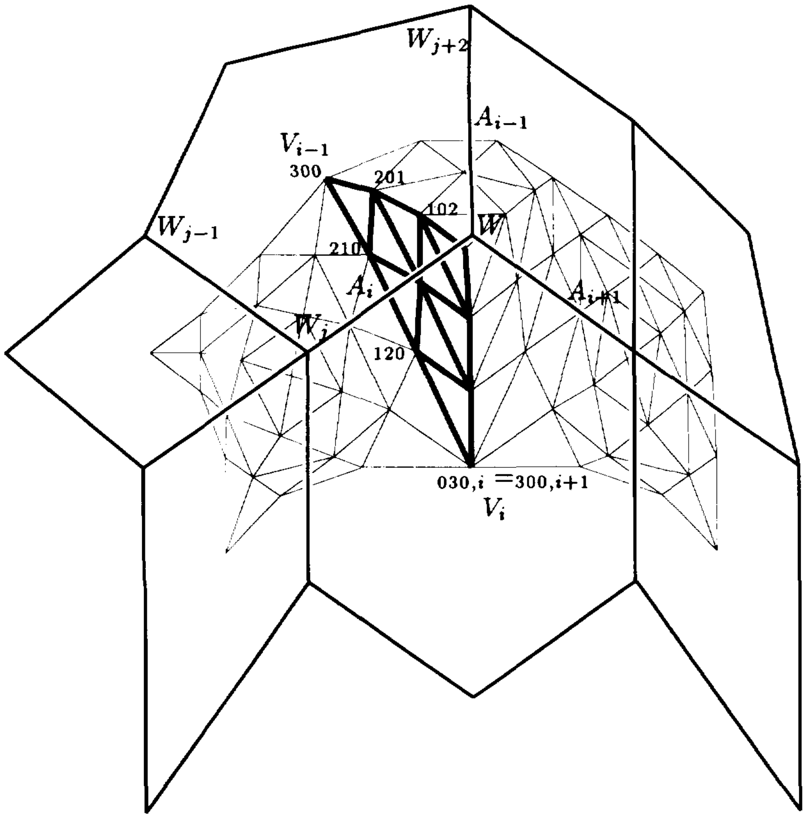


Fig. 1. Mesh of points W_l and the resulting quilt of three-sided patches. The control net and the labels of the control points of one patch are emphasized.

that at most four cells meet at any interior mesh point. Since the construction averages the mesh points, the planar cells should preferably be convex. Let e be the number of interior edges of the input mesh. The output of the algorithm are the Bernstein–Bézier coefficients $P_{ijk,l} \in \mathbb{R}^3$ of $2e$ three-sided cubic patches p_l,

$$p_l(u,v) := \sum_{i+j+k=3} P_{ijk,l} \frac{3!}{i!j!k!} u^i v^j w^k, \qquad u,v \in [0,1], w := 1-u-v,$$

(cf., Farin [1990]) of a C^1 surface that follows the outline of the input mesh. By default, a global surface boundary consists of the boundaries of the well-defined patches one removed from the global mesh boundary. This is illustrated in Figure 1 and later in Figure 3. Using this rule, we can match particular boundary data by adjusting or extending the mesh.

The algorithm has two steps:

(1) On each cell, choose a point V, and on each edge, choose a point A. For nonplanar (hence, quadrilateral) cells with vertices W_j, $j = 1 \ldots m$, V

should be the centroid, and each A_j an edge midpoint; that is,

$$V := \sum \frac{W_j}{m}, \qquad A_j := \frac{W_j + W_{j+1}}{2}, \qquad j = 1 \ldots m.$$

(2) Each edge $i = 1 \ldots n$, $n \in \{3, 4\}$ emanating from a mesh point W gives rise to a patch as follows: Let A_{i-1}, A_i, and A_{i+1} be the points on consecutive edges, and let V_{i-1} and V_i be the points in the faces that share A_i. Then the ith patch corresponding to the vertex has the following coefficients:

$$P_{300,i} = V_{i-1} \quad P_{210,i} = (2A_i + V_{i-1})3 \quad P_{120,i} = (2A_i + V_i)/3 \quad P_{030,i} = V_i$$

if W has	$n = 3$ neighbors	$n = 4$ neighbors
$P_{201,i} =$	$(P_{300,i} + P_{210,i} + P_{120,i-1})/3$	$(P_{210,i} + P_{120,i-1})/2$
$P_{111,i} =$	$(A_{i-1} + A_{i+1} + 5A_i + V_{i-1} + V_i, +3\ell(V_{i-1} + V_i - 2A_i))/9$	$(A_{i-1} + A_{i+1} + 4A_i +2\ell(V_{i-1} + V_i - 2A_i))/6$
$P_{021,i} =$	$(P_{030,i} + P_{120,i} + P_{210,i+1})/3$	$(P_{120,i} + P_{210,i+1})/2$
$P_{102,i} =$	$(P_{201,i} + P_{111,i} + P_{111,i-1})/3$	$(P_{111,i} + P_{111,i-1})/2$
$P_{012,i} =$	$(P_{021,i} + P_{111,i} + P_{111,i+1})/3$	$(P_{111,i} + P_{111,i+1})/2$

$$P_{003,i} = \frac{1}{n} \sum_{k=1}^{n} P_{012,k}$$

This completes the algorithm except for the specification of ℓ. Let Q be the patch abutting P, labeled so that $Q_{ij0} = P_{ij0}$. Then $\ell := l_0 - l_1$, where l_0 and l_1 (and the scalars m_0 and m_i) solve

$$\begin{aligned} l_0(P_{210} - P_{300}) &= m_0(P_{201} - P_{300}) + (1 - m_0)(Q_{201} - Q_{300}), \\ l_1(P_{030} - P_{120}) &= m_1(P_{021} - P_{120}) + (1 - m_1)(Q_{021} - Q_{120}). \end{aligned} \qquad (*)$$

That is, l_j is the relative length of the projection of two transversal tangents onto the versal tangent of the common boundary (see Fig. 2). For example, if the cell associated with V_{i-1} is an affine n-gon and the cell associated with V_i is an affine m-gon, then $2\ell = \cos(2\pi/n) + \cos(2\pi/m)$.

3. PROPERTIES OF THE SMOOTHED SURFACE

A number of properties of the smoothed surface are easy to establish. Since the vertex of each patch, $V = P_{300}$, is interpolated, the surface can interpolate one point on each planar cell. Since either a cell is planar or we choose its centroid and the edge midpoints, all A_i of a cell lie in the same plane, the tangent plane at V. Hence, all planar cells are tangent planes of the surface. If all of the coefficients $P_{ijk,l}$ are convex combinations of the mesh points, then the surface lies in the *convex hull* of the polyhedral mesh. We see that the former is the case if $\ell \in [-1/3, 5/6]$ for $n = 3$ and if $\ell \in [0, 1]$ for $n = 4$. If more is known about the structure of the mesh, the convex hull property can be established independent of ℓ (see the Remark below). It remains to prove the tangent plane continuity of the resulting surface.

THEOREM. *The piecewise cubic surface generated by the algorithm is* C^1.

PROOF. Using the abbreviation $[c_0, c_1, \dots, c_d]$ for the polynomial $\sum_{i=0}^{d} c_i(1-t)^{d-i}t^i$, we define the scalar polynomials a and b and the vector-valued polynomials $D_1 p$, $D_2 p$, and $D_2 q$ as follows:

$$n_0 := 1 - m_0, \qquad n_1 := 1 - m_1,$$

$$a := [1,1], \qquad b := \left[\frac{l_0}{n_0}, 2\left(\frac{l_0}{n_1} + \frac{l_1}{n_0}\right), \frac{l_1}{n_1}\right], \qquad c := -\left[\frac{m_0}{n_0}, \frac{m_1}{n_1}\right],$$

$$D_1 p := 2[A_i - V_{i-1}, V_i - A_i],$$

$$D_2 p := 3[P_{201,i} - P_{300,i}, 2(P_{111,i} - P_{210,i}), P_{021,i} - P_{120,i}],$$

$$D_2 q := 3[Q_{201,i} - Q_{300,i}, 2(Q_{111,i} - Q_{210,i}), Q_{021,i} - Q_{120,i}].$$

Here, $D_1 p$ is short for the versal derivative $(\partial p/\partial t)(t, 0)$ and expands to $2(A_i - V_{i-1})(1-t) + 2(V_i - A_i)t$. Similarly, $D_2 p$ and $D_2 q$ represent the transversal derivatives of the patches along the common boundary. It suffices to show that all three derivatives lie in a common plane; that is,

$$aD_2 q = bD_1 p + cD_2 p.$$

A cusping match of the patches is ruled out because $a(t)c(t) < 0$ (cf. Peters 1991, Lemma 2.1). Equating the four coefficients of the two cubic polynomials $aD_2 q$ and $bD_1 p + cD_2 p$ either side, we find that the first and the last coefficients agree due to eq. $(*)$, while the remaining two must be verified by lengthy but standard substitution of A_l and V_l, using $k_P = 1/3$ if $n = 3$ and $k_p = 1/2$ if $n = 4$, and the identities

$$P_{201,i} - P_{300,i} = 2k_P \frac{A_{i-1} + A_i - 2V_{i-1}}{3},$$

$$P_{021,i} - P_{120,i} = 2\frac{(1 - 2k_P)V_i + k_P A_{i+1} + (k_P - 1)A_i}{3}. \qquad \square$$

Remark (for the expert). The two equations associated with the second and third coefficient of $bD_1 p + cD_2 p - aD_2 q$ in the unknowns P_{111} and Q_{111} have a family of solutions parameterized by r (the formula for Q_{111} follows by symmetry):

$$3P_{111} := \frac{k_P}{r} A_{i-1} + k_P r A_{i+1} + \frac{(l_1 - 1 + k_P)r^2 + (2 - \ell)r + (k_P - l_0)}{r} A_i + \frac{(1 - l_1)r + l_0 - 2k_P}{r} V_{i-1} + ((1 - l_1 - 2k_P)r + l_0)V_i.$$

The algorithm uses $r = 1$ to maximize symmetry of the formulas. If more is known about the mesh, an asymmetric choice can be preferable. For example,

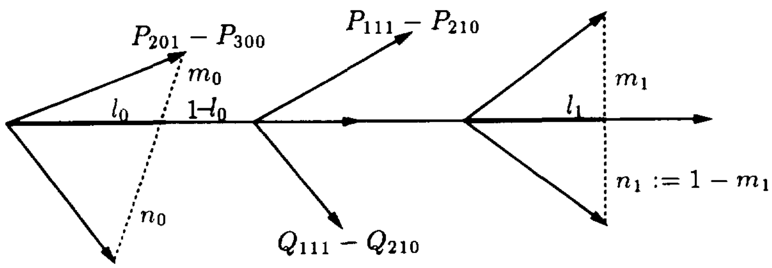


Fig. 2. Definition of the six scalars $l_0, l_1, m_0, m_1, n_0, n_1$.

in Peters [1995], the cell containing V_{i-1} is quadrilateral and the cell containing V_i is an affine m-gon. Therefore, $l_0 = 0$, and $\ell = -l_1 = \cos(2\pi/m)/2$, and the parameter r is chosen so that the convex hull property holds also for $m = 3$.

COROLLARY. *If $n = 4$ and $\ell = 0$, then the patch generated by the algorithm is quadratic.*

PROOF. Since the boundary curve is a degree-raised quadratic and

$$6P_{111} := A_{i-1} + A_{i+1} = 4A_i + 2\ell(V_{i-1} + V_i - 2A_i),$$

the construction yields a degree-raised quadratic polynomial patch with coefficients

$$\begin{array}{ccccc} & & \mathrm{P}_{002} & & \\ & \mathrm{P}_{101} & & \mathrm{P}_{011} & \\ \mathrm{P}_{200} & & \mathrm{P}_{110} & & \mathrm{P}_{020} \end{array} = \begin{array}{ccccc} & & \frac{1}{4}\sum \mathrm{A}_j & & \\ & \frac{A_{i-1} + A_i}{2} & & \frac{A_{i+1} + A_i}{2} & \\ \mathrm{V}_{i-1} & & \mathrm{A}_i & & \mathrm{V}_i. \end{array}$$

To check the formulas, the author implemented the algorithm and generated surfaces from meshes. Figure 3 shows a surface resulting from a vertex-degree-bounded polyhedral mesh that has a boundary. Figure 4 shows a closed surface, a Christmas ornament if you like. Figure 5 illustrates the conversion of an original mesh to a special type of vertex-degree-bounded polyhedral mesh. Here, the refinement has been chosen intentionally to round all features and to create a rounded object. A more functional smoothing of the gripper, using locally adjusted change of curvature, is shown in Peters [1995] or http://www.cs.purdue.edu/people/jorg.

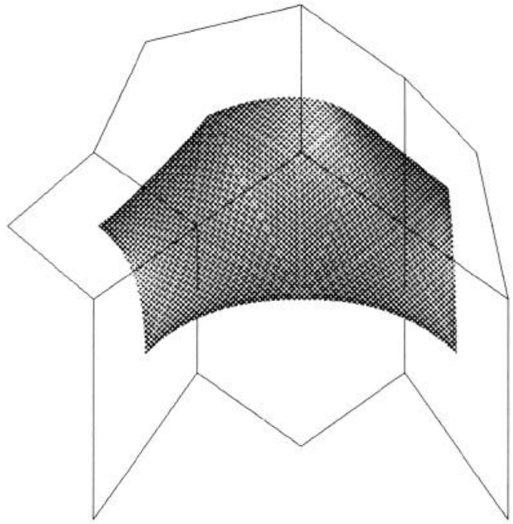


Fig. 3. Surface resulting from a vertex-degree-bounded polyhedral mesh with a boundary.

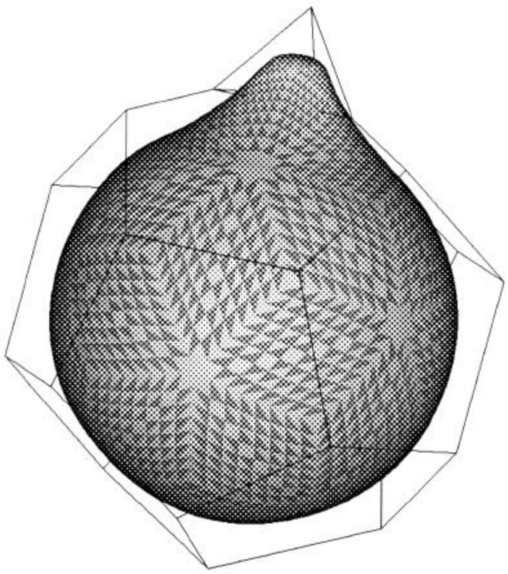
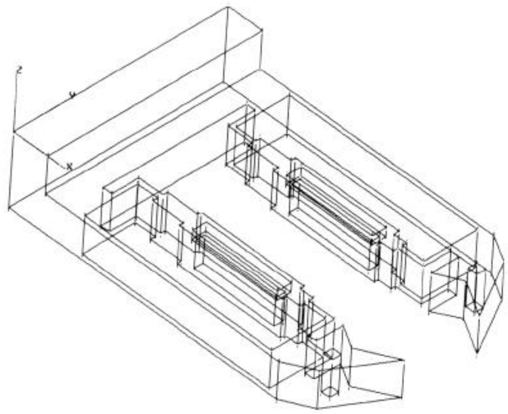
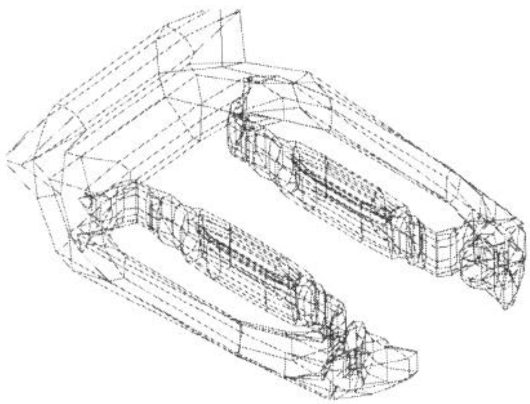


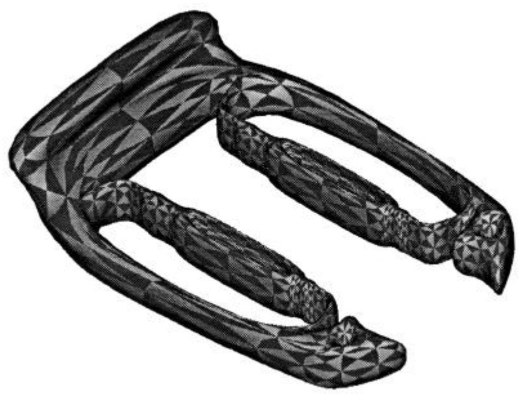
Fig. 4. Surface resulting from a vertex-degree bounded polyhedral mesh without boundary.



(a)



(b)



(c)

Fig. 5. (a) Original mesh. (b) Vertex-degree-bounded polyhedral mesh obtained from the Doo–Sabin algorithm with ratio 1/2. (c) Smooth surface.

REFERENCES

DE BOOR, C. W., HÖLLIG, K., AND RIEMENSCHNEIDER, S. 1994. *Box Splines.* Springer-Verlag, New York.

CATMULL, E., AND CLARK, J. 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. *Comput. Aided Des. 10*, 6 (Nov.), 350–355.

DOO, D. 1978. A subdivision algorithm for smoothing down irregularly shaped polyhedrons. In *Proceedings on Interactive Techniques in Computer Aided Design* (Bologna, Italy), pp. 157–165.

FARIN, G. 1990. Curves and surfaces for computer aided geometric design. Academic Press, New York.

LEE, S. L., AND MAJID, A. A. 1991. Closed smooth piecewise bicubic surfaces. *ACM Trans. Graph. 10*, 4 (Oct.), 352–365.

LOOP, C. T. 1987. Smooth subdivision surfaces based on triangles. Master's thesis, Dept. of Computer Sciences, Univ. of Utah, Salt Lake City.

LOOP, C. T. 1994. Smooth spline surfaces over irregular meshes. In *Proceedings of ACM SIGGRAPH.* (Orlando, FL, Aug. 1994)

MANN, S. 1992. Surface approximation using geometric Hermite patches. Thesis, Dept. of Computer Sciences, Univ. of Washington, Seattle.

PETERS, J. 1995. C^1 surface splines. *SIAM J. Numer. Anal. 32*, 2, 645–666.

PETERS, J. 1993. Smooth free-form surfaces over irregular meshes generalizing quadratic splines. (Sept.), *Comput. Aided Geom. Des. 10,* 5 347–361.

PETERS, J. 1992. Constructing C^1 surfaces of arbitrary topology using biquadratic and bicubic splines. In *Designing Fair Curves and Surfaces*, N. Sapidis, Ed., SIAM, Philadelphia, Pa.

PETERS, J. 1991. Smooth interpolation of a mesh of curves. *Constructive Approximation 7*, 221–247.

PIAH, A. R. M. 1991. Construction of smooth surfaces by piecewise tensor product polynomials, Rep. CS 91-04, Univ. of Dundee, Computer Sciences, Scotland, Feb.

PIPER, B. 1987. Visually smooth interpolation with triangular Bézier patches. In *Geometric Modelling: Algorithms and Trends*, G. E. Farin, Ed. SIAM, Philadelphia, Pa., pp. 221–233.

SHIRMAN, L. AND SÉQUIN, C. 1987. Local surface interpolation with Bézier patches. *Comput. Aided Geom. Des. 4*, 279–295. (Errata in *Comput. Aided Geom. Des. 8*, 3, 1991, 217–221).

SMALLEY, R. E. 1991. The third form of Carbon. *Nav. Res. Rev. 3.*

Received November 1993; revised July; accepted October 1994