Smoothing Polyhedra using Implicit Algebraic Splines*

Chandrajit L. Bajaj *Insung Ihm*

Department of Computer Sciences,
Purdue University,
West Lafayette, Indiana 47907

Tel: 317-494-6531
Fax: 317-494-0739
bajaj@cs.purdue.edu
ihm@cs.purdue.edu

Abstract

Polyhedron "smoothing" is an efficient construction scheme for generating complex boundary models of solid physical objects. This paper presents efficient algorithms for generating families of curved solid objects with boundary topology related to an input polyhedron. Individual faces of a polyhedron are replaced by low degree implicit algebraic surface patches with local support. These quintic patches replace the C^0 contacts of planar facets with C^1 continuity along all interpatch boundaries. Selection of suitable instances of implicit surfaces as well as local control of the individual surface patches are achieved via simultaneous interpolation and weighted least-squares approximation .

1 Introduction

The generation of a C^1 mesh of smooth surface patches or *splines* that interpolate or approximate *triangulated space data* is one of the central topics of geometric design. Two surfaces $f(x, y, z) = 0$ and $g(x, y, z) = 0$ meet with C^k-continuity along a curve C if and only if there exists functions $\alpha(x, y, z)$ and $\beta(x, y, z)$ such that all derivatives upto order k of $\alpha f - \beta g$ equals zero at all points along C, see for e.g., [13]. Chui [8], Dahmen and Michelli [10] and Hollig [18] summarize much of the history of multivariate splines. Prior work on splines have traditionally worked with a given planar triangulation using a polynomial function basis [1, 32, 35]. More recently surface fitting has been considered over closed triangulations in three dimensions using parametric surface patches [6, 7, 12, 15, 16, 17, 21, 24, 26, 27, 29, 33, 37].

*Supported in part by NSF grants CCR 90-00028, DMS 91-01424 and AFOSR contract 91-0276

Little work has been done on spline bases using implictly defined algebraic surface patches. Sederberg [34] showed how various smooth implicit algebriac surfaces in trivariate Bernstein basis can be manipulated as functions in Bezier control tetrahedra with finite weights. Patrikalakis and Kriezis [25] extended this by considering implicit algebriac surfaces in a tensor product B-spline basis. However the problem of selecting weights or specifying knot sequences for C^1 meshes of implicit algebraic surface patches which fit given spatial data, was left open. Dahmen [9] presented a scheme for constructing C^1 continuous, piecewise quadric surface patches over a data triangulation in space. In his construction each triangular face is split and replaced by six micro quadric triangular patches, similar to the splitting scheme of Powell-Sabin [30]. Dahmen's technique however works only if the original triangulation of the data set allows a transversal system of planes, and hence is quite restricted. Moore and Warren [23] extend the marching cubes scheme of [22] and compute a C^1 piecewise quadratic approximation (least-squares) to scattered data. They too use a Powell-Sabin like split, however over subcubes.

In this paper we consider an arbitrary spatial triangulation $\mathcal{T}$ consisting of vertices $\mathbf{p} = (x_i, y_i, z_i)$ in $\mathbb{R}^3$ (or more generally a simplicial polyhedron $\mathcal{P}$ when the triangulation is closed), with possibly "normal" vectors at the vertex points. We present an algorithm to construct a C^1 continuous mesh of low degree real algebraic surface patches S_i , which respects the topology of the triangulation $\mathcal{T}$ or simplicial polyhedron $\mathcal{P}$, and C^1 interpolates all the vertices. Our technique is compleletly general and uses a single implicit surface patch for each triangular face of $\mathcal{T}$ of $\mathcal{P}$, i.e. no local splitting of triangular faces. Furthermore, our C^1 interpolation scheme is local in that each triangular surface patch has independent degrees of freedom which may be used to provide local shape control. In this paper, we show how these extra parameters may be adjusted and the shape of the patch controlled by using weighted least squares approximation from additional points and normals, generated locally for each triangular patch.

Algebraic surfaces: For our polyhedron smoothing problem we only considered fitting with algebraic surfaces, i.e. two dimensional zero sets of polynomial equations. This was primarily motivated from the fact that manipulating polynomials, as opposed to arbitrary analytic functions, is computationally more efficient [3]. Furthermore, algebraic surfaces provide enough generality to accurately model almost all complicated rigid objects. A real algebraic surface S in $\mathbb{R}^3$ is implicitly defined by a single polynomial equation $\mathcal{F}$: $f(x, y, z) = 0$, where coefficients of f are over the real numbers $\mathbb{R}$. While all real algebraic surfaces have an implicit definition $\mathcal{F}$ only a small subset of these real surfaces can also be defined parametrically by the triple $\mathcal{G}(s,t)$: $(x = G_1(s,t), y = G_2(s,t), z = G_3(s,t))$ where each G_i, $i = 1, 2, 3$, is a rational function (ratio of polynomials) in s and t over $\mathbb{R}$. The primary advantage of the implicit definition $\mathcal{F}$ is its closure properties under modeling operations such as intersection, convolution, offset, blending, etc. The smaller class of parametrically defined algebraic surfaces $\mathcal{G}(s,t)$ is not closed under any of these operations. Closure under modeling operations allow cascading repetitions[1] without any need of approximation. Furthermore, designing with the complete class of algebraic surfaces leads to better possibilities (as we show here) of being able to satisfy the same geometric design constraints with much lower degree algebraic surfaces. The implicit representation of smooth algebraic surfaces also naturally yields half-spaces $\mathcal{F}^+ : f(x,y,z) \geq 0$ and $\mathcal{F}^- : f(x,y,z) \leq 0$, a fact quite useful for intersection and offset modeling operations. Finally, since prior approaches to scattered data fitting over triangulations had focused on the parametric representation of surfaces our aim here was to exhibit that implicitly defined algebraic surfaces were also equally (if not more) amenable to the task.

Why is low degree important ? Let the *geometric degree* of an algebraic surface is the maximum number of intersections between the surface and a line, counting complex, infinite and multiple intersections. It is a measure of the "wavi-ness" of the surface. This geometric degree is the same as the degree of the defining polynomial f of the algebraic surface in the implicit definition, but may be as high as n^2 for a parametrically defined surface with rational functions G_i of degree n. The *geometric degree* of an algebraic space curve is the maximum number of intersections between the curve and a plane, counting complex, infinite and multiple intersections. A well known theorem of algebraic geometry (Bezout's theorem) states that the the geometric degree of an algebraic intersection curve of two algebraic surfaces may be as large as the product of the geometric degrees of the two surfaces [36]. The use of low degree surface patches to construct models of physical objects thus results in faster computations for subsequent geometric model manipulation operations such as computer graphics display, animation, and physical object simulations, since the time complexity of these manipulations is a direct function of the degree of the involved curves and surfaces. Furthermore, the number of singularities[2] (sources of numerical ill-conditioning) of a curve of geometric degree m may be as high as m^2 [38]. Keeping the degree low of the curves and surfaces thus leads to potentially more robust numerical computations.

The main results of this paper are:

1. an efficient algorithm in sections 2, 3, 4 which computes C^1 smooth models of a convex polyhedron using degree 5 algebraic surface patches, and of an arbitrary polyhedron using *at most* degree 7 algebraic surface patches,

2. a numerically stable method in section 5 for the simultaneous C^1 interpolation and weighted least squares approximation used for both the selection of a smooth, single-sheeted solution surface as well as local shape control,

Both our solution surface degree bounds 5 and 7 are also significantly better than the geometric degree 18, parametric bicubic surface patch solutions for the same problem achieved by Peters [26]. Note that this comparison is only between the prior known fitting algorithm which did not additionally split the meshed data and had the best degree bound. Details on the implementation of our algorithms and illustrative examples are given in the section 6.1.

[1] The output of one operation acts as the input to another operation

[2] Points on the curve where all derivatives are zero

2 The Polyhedron Smoothing Algorithm

In this section, we present an outline of the algorithm to smooth a simple polyhedron $\mathcal{P}$ with C^1-continuous implicit algebraic surface patches.

Algorithm

1. Triangulate each of the nontriangular polygonal faces of the given polyhedron $\mathcal{P}$. Each face of $\mathcal{P}$ is a simple polygon which can be triangulated by adding non-intersecting inner diagonals[31]. See Figure 2.

2. Specify a single "normal" vector at each vertex of $\mathcal{P}$. This provides a single tangent plane for all patches which shall interpolate that vertex with C^1 continuity.

3. Next, construct a curvilinear wire frame by replacing each edge of $\mathcal{P}$ with a curve which C^1-interpolates the end points of the edge and the specified "normals". Any remaining degrees of freedom of the C^1 interpolatory curve are used to select a desired shape of the curve and indirectly thereby a desired shape of the smoothing surface patch. See Figure 2.

4. Specify normal vectors along each of the edge curves. This provides the tangent planes for the two incident patches which shall C^1 interpolate the edge curves. See Figure 3.

5. Finally, C^1-interpolate the three edge curves and curve normals of each face. The remaining degrees of freedom for each individual patch are consumed via weighted least squares approximation to achieve a suitably shaped single-sheeted algebraic surface patch. The resulting surface patches yield a globally C^1 smooth curved model for the given polyhedron. See Figures 3 and 6.

Details of each of the steps 2 to 5 of the algorithm for specific classes of polyhedra (convex, non-convex) together with explicit degrees of the required curves and surfaces are presented in subsequent sections. Steps 2 to 4 are detailed in section 3 and step 5 in sections 4.

3 Wireframe Construction

3.1 Choice of Vertex Normals

The single "normal" vector assigned to each vertex of the triangulated polyhedron $\mathcal{P}$ can be chosen independently and quite arbitrarily. However the relative directions of each adjacent vertex normal pair can affect the degree of the C^1 interpolating edge curve which replaces th e straight edges of $\mathcal{P}$. Let the two normal vectors at the two end points of an edge be called an *edge-normal-pair*. Certain relative directions of an edge-normal-pair induce an inflection point for any C^1 interpolating curve. Since conics do not have inflection points one is then forced to either switch to cubic curves at the least or to artificially split the edge. Splitting an edge in turn induces splitting of the triangular face of $\mathcal{P}$. In this section, we restrict ourselves to surface fitting without the splitting of any triangular faces of $\mathcal{P}$.

We first derive a necessary and sufficient condition for the relative directions of an edge-normal-pair to allow a singly connected C^1 conic interpolating curve. Here, the interpolation is *strict* in that the curve's normal at the vertex points and the prescribed vertex normal are in the same direction and not opposite. This restriction guarantees the construction wire frames which are free of cusp-like connections. In the following definitions and lemmas we make all of this more precise.

Definition 3.1 *Let $P_0 = (p_0, n_0)$ and $P_1 = (p_1, n_1)$ be an edge-normal-pair. A conic segment $S(P_0, P_1)$ is said to C^1-interpolate P_0 and P_1 if there exists a non-degenerate quadratic surface a(quadric) $F : ax^2 + by^2 + cz^2 + dxy + eyz + fzx + gx + hy + iz + j = 0$ such that*

- *$S(P_0, P_1)$ is a singly connected conic segment on F,*
- *p_0 and p_1 are the end points of $S(P_0, P_1)$,*
- *the gradient of $f(x, y, z) = 0$ at p_0 and p_1 have the same directions as n_0 and n_1, respectively. In other words $\nabla f(p_0) = \alpha n_0$ and $\nabla f(p_1) = \beta n_1$ for constants $\alpha, \beta > 0$.*

For a given point-normal pair $P = ((p_x, p_y, p_z), (n_x, n_y, n_z))$, we have $T_P(x, y, z) = n_x(x - p_x) + n_y(y - p_y)n_z(z - p_z) = 0$ as the equation of the tangent plane that passes through (p_x, p_y, p_z) and has a normal direction (n_x, n_y, n_z). The tangent plane $T_P(x, y, z) = 0$ divides space into a positive halfspace $\{(x, y, z) \in \mathbf{R}^3 | T_P(x, y, z) > 0\}$, and a negative halfspace $\{(x, y, z) \in \mathbf{R}^3 | T_P(x, y, z) < 0\}$. Note also, that for a surface $f(x, y, z)$ if $\nabla f(p_x, p_y, p_z) = \alpha(n_x, n_y, n_z)$ then $T_P = T_{((p_x, p_y, p_z), \nabla f(p_x, p_y, p_z))}$.

Theorem 3.1 *There exists a single connected conic segment $S(P_0, P_1)$ on a non-degenerate quadric F that C^1-interpolates $P_0 = (p_0, n_0)$ and $P_1 = (p_1, n_1)$ if and only if $T_{P_0}(p_1) \cdot T_{P_1}(p_0) > 0$.*

Proof: ($\Rightarrow$) Let $f(x, y, z) = ax^2 + by^2 + cz^2 + dxy + eyz + fzx + gx + hy + iz + j = 0$ be the non-degenerate quadric F. From definition 3.1 of a C^1 interpolating conic segment on F, it follows that $T_{P_0}(p_1) \cdot T_{P_1}(p_0) = T_{(p_0, \nabla f(p_0))}(p_1) \cdot T_{(p_1, \nabla f(p_1))}(p_0)$. Without loss of generality, assume that $p_0 = (0,0,0)$, and $p_1 = (1,0,0)$. Since $\nabla f(x, y, z) = (2ax + dy + fz + g, 2by + dx + ez + h, 2cz + ey + fx + i)$, $\nabla f(0,0,0) = (g, h, i)$ and $\nabla f(1,0,0) = (2a+g, d+h, f+i)$. Hence, $T_{(p_0, \nabla f(p_0))}(x, y, z) = gx + hy + iz$, and $T_{(p_1, \nabla f(p_1))}(x, y, z) = (2a+g)(x-1) + (d+h)y + (f+i)z$. From the containment conditions of the two points, $f(0,0,0) = j = 0$, and $f(1,0,0) = a + g + j = a + g = 0$. Then, $T_{(p_0, \nabla f(p_0))}(p_1) \cdot T_{(p_1, \nabla f(p_1))}(p_0) = g(-(2a+g)) = -g(2(-g)+g) = g^2 > 0$, as g cannot be zero. For if $g = 0$, it would follow that $a = g = j = 0$, and either $T_{P_0}(p_1)$ or $T_{P_1}(p_0)$ or both would be zero (i.e. the tangent plane at p_0 contains p_1 or the tangent plane at p_1 contains p_0, both). In each such case the quadric $f(x, y, z) = by^2 + cz^2 + dxy + eyz + fzx + hy + iz = 0$ is a degenerate quadric as its intersection with any plane section through p_0 and p_1 yields a pair of lines (a degenerate conic).
($\Leftarrow$) If $T_{P_0}(p_1) \cdot T_{P_1}(p_0) > 0$, then the conic segment on $f(x, y, z) = L(x, y, z)^2 - \kappa \cdot T_{P_0}(x, y, z) \cdot T_{P_1}(x, y, z) = 0$ or $-f(x, y, z) = 0$ will C^1 interpolate the pair P_0 and P_1, where $L(x, y, z) = 0$ is a plane containing p_0 and p_1, and κ is a constant. ♠

The geometric interpretation of the inequality $T_{(p_0, \nabla f(p_0))}(p_1) \cdot T_{(p_1, \nabla f(p_1))}(p_0) > 0$ is that p_0 is on the positive (negative) halfspace of T_{P_1} if and only if p_1 is on the positive (negative) halfspace of T_{P_0}.

3.2 Generation of a Conic Wireframe

First, we give a definition of the term *quadric wire*.

Definition 3.2 *Let $C(t) = (\frac{x(t)}{w(t)}, \frac{y(t)}{w(t)}, \frac{z(t)}{w(t)})$ and $N(t) = (\frac{nx(t)}{w(t)}, \frac{ny(t)}{w(t)}, \frac{nz(t)}{w(t)})$ be two triples of quadratic rational parametric polynomials. Then, the pair $W(t) = (C(t), N(t))$ is called a quadric wire if there exists a quadric $q(x, y, z) = 0$ such that $q(C(t)) = 0$ and $\nabla q(C(t)) = \alpha N(t)$, for $\alpha > 0$ and all t.*

The rationale in our construction of a quadric wire is that a conic curve is naturally associated with curve normal vectors taken from a quadric. Our first step to smoothing a convex polyhedron is to compute a C^1 interpolating conic curve $C(t)$, from an edge-normal-pair (p_0, n_0), (p_1, n_1) and a normal npl of a plane Q which contains p_0 and p_1. In particular, we set $W(0) \equiv (p_0, n_1)$ and $W(1) \equiv (p_1, n_1)$,[3] and hence use the segment of $W(t), 0 \leq t \leq 1$. To compute $C(t)$, the normal vectors n_0 and n_1 are projected into the plane P on which $C(t)$ will lie. (See Figure 1). This projection results in a control triangle $p_0 - p_2 - p_1$. Lee [20] presents a compact method for computing a conic curve $C(t)$ from such a control triangle. In his formulation, the conic is expressed in Bernstein-Bézier form :

$$C(t) = \frac{w_0 p_0 (1-t)^2 + 2 w_2 p_2 t(1-t) + w_1 p_1 t^2}{w_0 (1-t)^2 + 2 w_2 t(1-t) + w_1 t^2},$$

where $w_i > 0$, $i = 0, 1, 2$ are shape control parameters. An often used parameterization, called the *rho-conic parameterization*, is

[3] By $\equiv$, we mean the points are the same, and the normal vectors are proportional, maintaining positivity.

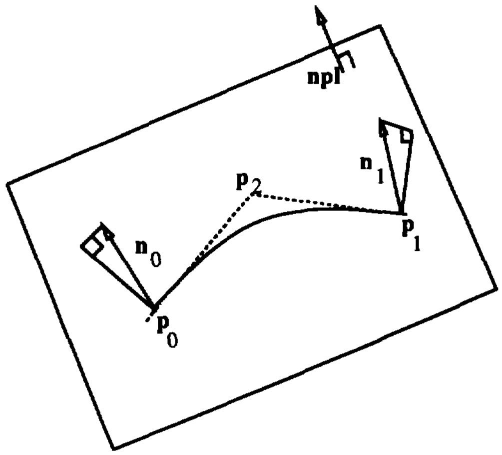

Figure 1: Computation of a Conic Curve

given by the special choice $w_0 = w_1 = (1-\rho)$, $w_2 = \rho$, $\rho > 0$. Let $p_{01} = (p_0 + p_1)/2$ be the midpoint of the chord p_0p_1. Then, ρ has a property that $C(0.5) - p_{01} = \rho(p_2 - p_{01})$. From this, we can see that as ρ is increased, the conic gets more curved. In particular, it can be shown that $\rho = 0.5$ for a parabola, $0 < \rho < 0.5$ for ellipses and $0.5 < \rho < 1.0$ for hyperbolas.

3.3 Assigning Normals along Edge curves

Once $C(t)$ is fixed, we find a quadratic surface $q(x, y, z) = 0$ such that $N(t)$ is proportional to $\nabla q(x, y, z)$ along $C(t)$ and interpolates n_0 and n_1. Consider a quadric surface $q(x, y, z) = c_0x^2 + c_1y^2 + c_2z^2 + c_3xy + c_4yz + c_5zx + c_6x + c_7y + c_8z + c_9 = 0$. $q(x, y, z) = 0$ has 10 coefficients, and since dividing the surface by any nonzero coefficient does not change the surface, there are 9 degrees of freedom. The first requirement is that $q(x, y, z) = 0$ must contain the computed conic $C(t)$. The C^1 interpolation algorithm for algebraic surfaces [5] gives 5 linear equations in terms of the unknowns c_i for the containment requirement. That 5 constraints on c_i are required also follows from Bezout's theorem which says if a non-degenerate conic intersects with a quadric at more than 4 points, then the conic must lie on the quadric.

Hence, 4 ($= 9 - 5$) degrees of freedom in choosing c_i are left, and these are used to interpolate the normal vectors at the two end points. Interpolating n_0 and n_1 at p_0 and p_1, respectively, gives 2 more linear constraints which leaves 2 degrees of freedom in choosing the quadric. We now explain why specifying one more normal vector at a point on the C^1 interpolating conic fixes the normal vectors along the entire conic. Consider the gradient vector $\nabla q(x, y, z)$ of the quadric. Its components are linear and the vector function $\nabla q(C(t))$ is a degree 2 polynomial parametric curve in projective space. Hence, three independent constraints fixes the curve $\nabla q(C(t))$ and thereby the normal vector along $C(t)$.

Using similar reasons as above one obtains the following lemma.

Lemma 3.1 *Let $W(t) = (C(t), N(t))$ be a quadric wire. Then the quadrics which C^1 interpolate $W(t)$ comprises of a family of surfaces with one degree of freedom.*

What we do in our implementation in order to fix the additional normal vector on the conic is the following. First, the average $n_{01} = (n_0 + n_1)/2$ is computed, and then n_{01} is projected onto the plane which contains $C(t)$. (All conics are planar). Next we require that the projected vector be perpendicular to the tangent at $C(0.5)$ i.e. the vector $C'(0.5)$. This then fixes all the normal vectors $N(t)$ along $C(t)$.

For a convex polyhedron, we can always specify a normal vector at each vertex such that the condition in Theorem 3.1 for each edge-normal-pair is satisfied. (For example, the average of normal vectors of incident faces of a vertex is one possible choice.) This implies that we can always construct a wireframe for a convex polyhedron whose curves and associated normal vectors are described in terms of quadric rational polynomials. Whether we can construct a similar conic wireframe for non-convex polyhedra is currently unresolved.

3.4 Generation of a Cubic Wireframe

The construction of a cubic wireframe follows along very similar lines as the conic wireframe construction. Each edge is now replaced by a polynomial parametric cubic curve, C^1 interpolating the vertex-normal pairs of the edge. Here no restrictions are imposed on the vertex-normal pairs as was the case for the conic wireframe of the earlier section. The construction of this cubic wireframe or cubic mesh of curves, see for example [11], is what has been used in the past and previously reported for example in [26]. We therefore omit further discussion of this construction and refer the reader to the earlier references.

4 Local Patch Generation

4.1 C^1 Interpolation of a Quadric Triangle

Definition 4.1 *An augmented triangle is an 9-tuple $T = (p_0, p_1, p_2, n_0, n_1, n_2, npl_{01}, npl_{12}, npl_{20})$ where the points p_i are three vertices of a triangle with the corresponding unit normal vectors n_i, and npl_{ij} is the normal of the plane which will contain the quadric wire made from (p_i, n_i) and (p_j, n_j).*

Definition 4.2 *A quadric triangle is a triple $QT = (W_0(t), W_1(t), W_2(t))$ of quadric wires such that $W_0(1) \equiv W_1(0)$, $W_1(1) \equiv W_2(0)$, and $W_2(1) \equiv W_0(0)$.*

Given an augmented triangle, each quadric wire is computed as described in the foregoing section. Next the quadric triangle is fleshed using a single algebraic surface $f(x, y, z) = 0$. For this we use the C^1 interpolation of Bajaj and Ihm [5]. This algorithm takes as input positional and first derivative information of points and space curves, given parametrically or implicitly, and characterizes, in terms of the nullspace of a matrix, the space of all the algebraic surfaces of a specified degree that C^1 interpolates the specified geometric data. For the quadric triangle the C^1 interpolation is applied to all three quadric wires and produces a homogeneous linear system $\mathbf{M}_1\mathbf{x} = \mathbf{0}$, where unknowns $\mathbf{x}$ are coefficients of $f(x, y, z) = 0$, such that any algebraic surface with coefficients that are solutions of the system C^1 interpolates the quadric triangle. The nontrivial solutions in the nullspace of $\mathbf{M_I}$ form a family of all possible algebraic surfaces of degree n, satisfying the given input constraints, whose coefficients are expressed by homogeneous combinations of q free parameters where $q = n_v - r$ is the dimension of the nullspace. Since dividing $f(x, y, z) = 0$ by a nonzero number does not change the surface, there are, in fact, $n_v - r - 1$ degrees of freedom in choosing an instance surface from the family. Hence, the rank r of $\mathbf{M_I}$ must be less than the number of the coefficients n_v, should there exist an interpolating surface.

We now derive general degree bounds for C^1 interpolatory triangular patches with degree m interpolatory curves and from this obtain lower bounds on the degree of surfaces which C^1 interpolate a quadric triangle. Assume that we use a degree n algebraic surface $f(x, y, z) = 0$ to C^1 interpolate a wire of degree m $W(t) = (C(t), N(t))$. According to Bezout's theorem, $mn + 1$ constraints on the coefficients of f are required for the algebraic surface f to contain $C(t)$ which is of degree m. Additionally for C^1 continuity, consider the restricted normal vector $\nabla f(C(t))$. Since the degree of each component of $\nabla f(x, y, z)$ is, at most $n - 1$, each component of $\nabla f(C(t))$ has degree $m(n - 1)$. Furthermore, the vector function $\nabla f(C(t)) - \alpha N(t)$ is a degree $m(n-1)$ parametric polynomial curve in projective space, with $N(t)$ of degree m and α any polynomial of degree at most $m(n - 2)$. Finally, since the surface $f(x, y, z) = 0$ contains $C(t)$ the component of the above vector function along the tangent direction of $C(t)$ is already satisfied. Hence $m(n - 1) + 1$ additional constraints are enough to guarantee C^1 continuity along $C(t)$.

Lemma 4.1 *Let $W(t) = (C(t), N(t))$ be a degree m wire. For an algebraic surface $f(x, y, z) = 0$ of degree n to smoothly interpolate $W(t)$, at most $2mn - m + 2$ $(= mn + 1 + m(n - 1) + 1)$ independent linear constraints on the f's coefficients must be satisfied.*

This lemma says that the rank of the matrix for Hermite interpolation of a degree m wire with a degree n surface is at most $2mn - m + 2$. For C^1 interpolation of a triangular patch, there exists a geometric dependency between the three wires which also leads to dependency amongst these linear C^1 contraints. First, since the curves intersect pairwise, there must be three rank deficiencies between the equations from the containment conditions (i.e. three equations are generated twice). For the same reasons there must be three rank deficiencies between the equations for the matching of normals. Secondly, at each vertex of the curvilinear triangle, two incident curves automatically determine the normal at the vertex. It is obvious, from the way the curve wire construction, this vector is proportional to the given unit normal vector at the vertex. So, satisfying the containment conditions for the 3 curves guarantees that any interpolating surface has gradient vectors at the three points as required. This fact implies that there are three rank deficiencies between the linear equations for the containment conditions, and the equations for the C^1 condition. This yields a total of 9 overall deficiencies.

Lemma 4.2 *Let $QT = (W_0(t), W_1(t), W_2(t))$ be a quadric triangle. The rank of the linear system $\mathbf{M_I x} = \mathbf{0}$ which is constructed by the Hermite interpolation for the algebraic surface $f(x, y, z) = 0$ of degree n that smoothly fleshes QT, is at most $12n - 9$.*

Proof. For C^1 interpolation of all three quadric wires, $3(4n - 2 + 2) = 12n$ linear equations are generated according to Lemma 4.1. Subtracting 9 deficiencies from this yields $12n - 9$. ♠

Since $f(x, y, z) = 0$ of degree n has $\binom{n+3}{3}$ coefficients, and the rank of the linear system should be less than the number of coefficients for a nontrivial surface to exist, we see that 5 is the minimum degree required. In the quintic case, there are 56 coefficients (55 degrees of freedom) and the rank is at most 51, which results in a family of interpolating surfaces with at least 4 degrees of freedom in selecting an instance surface from the family.

Even though some special combination of three quadric wires can be interpolated by a surface of degree less than 5, for example, three quadric wires from a sphere, the probability that such spatial dependency occurs, given an arbitrary triple of conics with normals, is infinitesimal. Hence, we can say that 5 is the minimum degree required *with the probability one*.

Lemma 4.3 *Let $QT = (W_0(t), W_1(t), W_2(t))$ be a cubic triangle whose wires are cubic rational polynomials. The rank of the linear system $\mathbf{M_I x} = \mathbf{0}$ which is constructed by the Hermite interpolation for the algebraic surface $f(x, y, z) = 0$ of degree n that smoothly fleshes QT, is at most $18n - 12$.*

Proof. For C^1 interpolation of all three degree 3 wires, $3(6n - 3 + 2) = 18n - 3$ linear equations are generated according to Lemma 4.1. Subtracting 9 deficiencies, yields $18n - 12$. ♠

The minimum degree of the C^1 interpolating surface is 7. In the quintic case, there are 120 coefficients (119 degrees of freedom) and the rank is at most 114, which results in a family of interpolating surfaces with at least 5 degrees of freedom in selecting an instance surface from the family.

Lemma 4.4 *Let $QT = (W_0(t), W_1(t), W_2(t))$ be a quadric triangle with one edge a cubic wire. The rank of the linear system $\mathbf{M_I x} = \mathbf{0}$ which is constructed by the Hermite interpolation for the algebraic surface $f(x, y, z) = 0$ of degree n that smoothly fleshes QT, is at most $14n - 10$.*

Proof. For C^1 interpolation of two quadric wires and a cubic wire, $2(4n - 2 + 2) + (6n - 3 + 2) = 14n - 1$ linear equations are generated according to Lemma 4.1. Subtracting 9 deficiencies from this yields $14n - 10$. ♠

The minimum degree of the C^1 interpolating surface is 6. In the degree 6 case, there are 84 coefficients (83 degrees of freedom) and the rank is at most 74, which results in a family of interpolating surfaces with at least 9 degrees of freedom in selecting an instance surface from the family.

Lemma 4.5 *Let $QT = (W_0(t), W_1(t), W_2(t))$ be a cubic triangle with one edge a quadric wire. The rank of the linear system $\mathbf{M_I x} = \mathbf{0}$ which is constructed by the Hermite interpolation for the algebraic surface $f(x, y, z) = 0$ of degree n that smoothly fleshes QT, is at most $16n - 11$.*

Proof: For C^1 interpolation of two cubic wires and a quadric wire, $4n - 2 + 2) + 2(6n - 3 + 2) = 16n - 2$ linear equations are generated according to Lemma 4.1. Subtracting 9 deficiencies from this yields $16n - 11$. ♠

The minimum degree of the C^1 interpolating surface is 7. In the degree 7 case, there are 120 coefficients (119 degrees of freedom) and the rank is at most 101, which results in a family of interpolating surfaces with at least 18 degrees of freedom in selecting an instance surface from the family.

4.2 Surface Selection and Local Shape Control

The result of a C^1 interpolation of a quadric triangle QT is a family of degree 5 algebraic surfaces $f(x, y, z) = 0$ with at least 4 degrees of freedom. Similarly C^1 interpolation of a cubic triangle is achieved with a 5 parameter family of degree 7 surfaces. These families are expressed as a linear combination of the nontrivial coefficients vectors in the nullspace of $\mathbf{M_I}$. To select a degree 5 or 7 surface from their respective families, values must be specified for these extra degrees of freedom.

We now show how weighted least squares approximation to additional points around the triangular patch, can be used for both selecting a suitable non-singular surface from the family as well as as local shape control. Let $S_0 = \{v_i \in \mathbf{R}^3 | i = 1, \cdots, l\}$ be a set of points which approximately describes a desirable surface patch. (These points can be selected for example from a sphere, paraboloid etc., centered around the curvilinear triangle). A linear system $\mathbf{M_A x} = 0$, where each row of $\mathbf{M_A}$ is constructed from the linear conditions $f(v_i) = 0$ with $\mathbf{x}$ containing the undetermined coefficients of the family. Conventional least squares approximation is to minimize $\| \mathbf{M_A x} \|^2$ over the nullspace of $\mathbf{M_I}$. Though minimizing $\| \mathbf{M_A x} \|^2$ does yield a good distance approximation it does not prevent the resulting surface from self-intersecting, pinching or splitting inside the triangle.

To rid our solution surfaces of such singularities and provide more geometric control , we instead approximate a monotonic trivariate function $w = f(x, y, z)$ rather than just the implicit surface $f(x, y, z) = 0$, the zero contour of the function. We first generate $S_0 = \{(v_i, n_i) | i = 1, \cdots, l\}$ where v_i are approximating points, and n_i are approximating gradient vectors at v_i. Then, from this set, we construct two more sets $S_1 = \{u_i | u_i = v_i + \alpha n_i, i = 1, \cdots, l\}$, and $S_{-1} = \{w_i | w_i = v_i - \alpha n_i, i = 1, \cdots, l\}$ for some small $\alpha > 0$. Next we set up the least squares system $\mathbf{M_A} = \mathbf{b}$ from the following three kinds of equations : $f(v_i) = 0$, $f(u_i) = 1$, and $f(w_i) = -1$. These equations give an approximating contour level structure of the function $w = f(x, y, z)$ near the inside of a quadric triangle. We found out that forcing well behaved contour levels rids the selected surfaces of self-intersection in the spatial region enclosed by the points. See Figures 6, 8, 9 and 10.

4.3 Compatibility and Non-Singularity Constraints

In this subsection, we briefly discuss why quintic surfaces which C^1-interpolate quadric triangles may be singular at the end vertices. Ihm [19] gives a theorem which presents a necessary regularity condition on C^1 interpolating surfaces.

Theorem 4.1 *Let $C_1(u)$ and $C_2(v)$ be two parametric curves with parametric normal directions $N_1(u)$ and $N_2(v)$ such that $C_1(0) = C_2(0) = p$, and that $N_1(0)$ and $N_2(0)$ are proportional. Then, any surface S, which interpolates the curves with tangent plane continuity, is singular at p unless $\frac{\langle N_1'(0), C_2'(0)\rangle}{\|N_1(0)\|} = \frac{\langle C_1'(0), N_2'(0)\rangle}{\|N_2(0)\|}$.*

The above theorem implies that enforcing two curves to have the same normal vectors at intersection points, does not guarantee the regularity of an interpolating surface at those points. The equation in the theorem is a necessary condition for regularity, indicating that, if the given curves and their normals do not satisfy the equation, any smoothly interpolating surface must be singular at p. In most cases, the above condition is not met when quadric triangles are constructed, and hence we observe singularities at the vertices. A good side-effect of these vertex singularities is that the vertex enclosure problem is automatically resolved.

This issue has been also addressed in the literature of parametric surface fitting. Peters [27] showed that not every mesh of parametric curves with well-defined tangent planes at the mesh points can be interpolated by smooth regularly parametrized surfaces with one surface patch per mesh face (also known as the vertex enclosure problem). In [28], he used singularly parametrized surfaces to enclose a mesh points when mesh curves emanating from the point do not satisfy a constraint, called the vertex enclosure constraint.

5 Computational Details and Examples

5.1 Solution of Interpolation and Least-Squares Matrices

For an algebraic surface $S : f(x, y, z) = 0$ of degree n, the C^1 interpolation conditions of section 4.1 produces a homogeneous linear system $\mathbf{M_I x} = 0$, $\mathbf{M_I} \in \mathbf{R}^{n_i \times n_v}$ of n_i equations and n_v unknowns where $\mathbf{x}$ is a vector of the $n_v (= \binom{n+3}{3})$ coefficients of S. A matrix $\mathbf{M_A} \in \mathbf{R}^{n_a \times n_v}$ for least-squares approximation is next constructed, similar to the construction of $\mathbf{M_I}$, for the additional points generated around the triangular patch as described in section 4.2.

For the case of quintic algebraic surface patches we solve the following, simultaneous interpolation and weighted least-squares approximation problem below. The case of other low degree (6 or 7) C^1 algebraic surfaces is nearly identical, with only modified sizes of the matrices.

$$\begin{array}{ll} \textit{minimize} & \| \mathbf{M_A x} - \mathbf{b} \|^2 \\ \textit{subject to} & \mathbf{M_I x} = 0, \end{array}$$

where $\mathbf{M_I} \in \mathbf{R}^{n_i \times 56}$ is a Hermite interpolation matrix, and $\mathbf{M_A} \in \mathbf{R}^{n_a \times 56}$ and $b \in \mathbf{R}^{n_a}$ are matrix and vector, respectively, for contour level approximation, and $\mathbf{x} \in \mathbf{R}^{56}$ is a vector containing coefficients of a quintic algebraic surface $f(x, y, z) = 0$.

To find the nullspace of $\mathbf{M_I}$ in a computationally stable manner, the singular value decomposition (SVD) of $\mathbf{M_I}$ is computed [14] where $\mathbf{M_I}$ is decomposed as $\mathbf{M_I} = U \Sigma V^T$ where

$U \in \mathbf{R}^{n_i \times n_i}$ and $V \in \mathbf{R}^{56\times 56}$ are orthonormal matrices, and $\Sigma = diag(\sigma_1, \sigma_2, \cdots, \sigma_s) \in \mathbf{R}^{n_i \times 56}$ is a diagonal matrix with diagonal elements $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_s \geq 0$ $(s = min\{n_i, 56\})$. It is known that the rank r of $\mathbf{M_I}$ is the number of the positive diagonal elements of Σ, and that the last $56 - r$ columns of V span the nullspace of $\mathbf{M_I}$. Hence, the nullspace of $\mathbf{M_I}$ is expressed as : $\{\mathbf{x} \in \mathbf{R}^{56} \mid \mathbf{x} = \sum_{i=1}^{56-r} w_i \mathbf{v}_{r+i},\ where\ w_i \in \mathbf{R},\ and\ \mathbf{v}_j\ is\ the\ jth\ column\ of\ V\}$, or $\mathbf{x} = V_{56-r}\mathbf{w}$ where $V_{56-r} \in \mathbf{R}^{56\times(56-r)}$ is made of the last $56 - r$ columns of V, and $\mathbf{w}$ a $(56 - r)$-vector. [4] $\mathbf{x} = V_{56-r}\mathbf{w}$ compactly expresses all the quintic surfaces which Hermite-interpolate the three quadric wires.

After substitution for $\mathbf{x}$, we lead to $\| \mathbf{M_A}\mathbf{x} - \mathbf{b} \| = \| \mathbf{M_A} V_{56-r}\mathbf{w} - \mathbf{b} \|$. Then, an orthogonal matrix $Q \in \mathbf{R}^{n_a \times n_a}$ is computed such that

$$Q^T \mathbf{M_A} V_{56-r} = R = \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$$

where $R_1 \in \mathbf{R}^{(56-r)\times(56-r)}$ is upper triangular. (This factorization is called a *Q-R factorization* [14]). Now, let

$$Q^T b = \begin{pmatrix} c \\ d \end{pmatrix}$$

where c is the first $56 - r$ elements. Then, $\| \mathbf{M_A} V_{56-r}\mathbf{w} - \mathbf{b} \|^2 = \| Q^T\mathbf{M_A} V_{56-r}\mathbf{w} - Q^T\mathbf{b} \|^2 = \| R_1\mathbf{w} - c \|^2 + \| d \|^2$. The solution $\mathbf{w}$ can be computed by solving $R_1\mathbf{w} = c$, from which the final fitting surface is obtained as $\mathbf{x} = V_{56-r}\mathbf{w}$.

5.2 Examples

In prior sections, we described how to compute low degree triangular algebraic surface patches from a given augmented curvilinear triangle. A polyhedron is smoothed by replacing its faces with the triangular patches meeting each other with tangent plane continuity. For the augmented triangles $T = (p_0, p_1, p_2, n_0, n_1, n_2, npl_{01}, npl_{12}, npl_{20})$ of the faces of a polyhedron, the normal data, i.e., three vertex normals and three edge normals, must be provided as well as the given three vertices. In some applications, the normal data may come with a solid, but, in general, only vertices and their facial information are provided.

The vertex normal $\mathbf{n}_i$ at each vertex $\mathbf{p}_i$ can be computed by averaging the normals of the faces incident to the vertex. Other assignment schemes which rely on the normals arsiing form a sphere or a paraboloid are also possible. For a convex triangulation $\mathcal{T}$ or polyhedron $\mathcal{P}$, the above choice of normals at vertices always yields compatible vertex-normal pairs (as per section 3) for C^1 conic interpolation and hence degree five surface patches suffice by results in section 4. However the above simplistic choice of vertex normals may yield incompatible vertex-normal pairs for a non-convex triangulation or polyhedron. To come up with a compatible vertex normal assignment for the non-convex case is an open problem. For now, we use a C^1 interpolating cubic curve whenever an incompatible vertex-normal pair arises, as in the non-convex case. Hence in this case we may need to use algebraic surface patches of degree 7, (as per section 4). Also, we average the normals of the faces incident to each edge (p_i, p_j), and take its cross product with the vector $p_j - p_i$ to get the edge normal vector npl_{ij}. After the normal data is computed, quadric wires are generated for the ρ value which is interactively controlled by the user.

[4] As mentioned before, in most cases, the rank r of $\mathbf{M_I}$ is 51. However, we keep the variable r because it is possible that there are more dependency between boundary curves and normal vectors though the chances are rare.

Example 5.1 *Construction of Quadric Wire Frames*

Figures 5 and 7 show two quadric wire frames for the same convex polyhedron[5] with the ρ values 0.4 (yielding ellipses) and 0.75 (yielding hyperbolas), respectively. □

Example 5.2 *Polyhedra Smoothed with Quintic Algebraic Surfaces*

Each of 32 faces of the polyhedron in Example 5.1 is replaced by a quintic implicit algebraic surface which smoothly fleshes its quadric triangle. Figures 6 and 8 respectively illustrate the C^1 surface meshes of $\rho = 0.4$ and 0.75. □

6 Remarks and Open Problems

6.1 Implementation Issues

We have presented a method that smooths out a polyhedron with C^1 continuous triangular algebraic surface patches. The polyhedron smoothing algorithms have been implemented in our distributed and collaborative geometric design environment SHASTRA [2], currently consisting of independent toolkit processes SHILP, GANITH and VAIDAK. For polyhedron smoothing, SHILP takes as input a polyhedron $\mathcal{P}$ and a user specified ρ value (for shape control), and computes quadric wires (if the normal condition is satisfied for the edge) or cubic wires. Next, for each triangular facet of curves, a GANITH computation is invoked via inter process communication and the facet C^1 fitted with a low degree (5 to 7) algebraic surface patch. Potentially, a separate GANITH process can be invoked for each individual facet on a network of workstations, to achieve maximal distributed parallelism. See Figure 10.

6.2 Open Problems

A number of open problems do remain. First, we need to devise a more robust way of generating the points and contour levels for the least squares approximation of section 4.2. While the heuristics for weighted least square approximation usually work well, sometimes we need to manually change, for example, the value of α in S_1 and S_{-1}. Secondly, we continue to work on smoothing an arbitrary polyhedron. We feel that quintic algebraic surfaces are also flexible enough for generating C^1 smooth nonconvex triangular surface patches. In this paper, we have shown that degree seven algebraic surfaces are sufficient, however not necessary. (See Figure 9 for a

[5] This polyhedron is gyroelongated triangular bicupola with its rectangular faces triangulated.

C^1 mesh of quintic surface patches over a nonconvex combination of quadric wires.) An open problem is to construct a wire frame for a non-convex polyhedron with conic curves. In this paper, we have shown that cubic wires are sufficient for the non-convex case, however they are not shown to be necessary. On approach to accomodate incompatible adjacent normals in the non-convex case is to subdivide edges into sub-edges and thereby faces into subfaces. We are currently exploring this approach.

Our ultimate goal is to construct arbitrary curved solids with the lowest algebraic degree surface patches, and to manipulate them through geometric operations such as boolean set operations. This ability will provide a geometric modeling system with a complex way of creating and manipulating models of physical objects with various geometries. One current application of our polyhedron smoothing algorithms has been in the smooth reconstruction of skeletal structures from three dimensional CT/NMR imaging data, using the SHILP, GANITH and VAIDAK toolkits of our SHASTRA system [2]. See also [4] for algorithmic details of the skeletal model reconstruction via approximation of the imaging data, using relatively sparse number of curved patches.

Acknowledgements : We are grateful to Vinod Anupam, Andrew Royappa and Dan Schikore for their assistance in the implementation of the smoothing algorithms.

References

[1] Alfeld, P.,. Scattered Data Interpolation in Three or More Variables. In T. Lyche and L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design*, pages 1–34. Academic Press, 1989.

[2] Anupam, A., and Bajaj, C., and Royappa, A. *The SHASTRA Distributed and Collaborative Geometric Design Environment*. Computer Science Technical Report, CAPO-91-38, Purdue University, 1991.

[3] Bajaj, C. Geometric modeling with algebraic surfaces. In D. Handscomb, editor, *The Mathematics of Surfaces III*, pages 3–48. Oxford Univ. Press, 1988.

[4] Bajaj, C. Electronic Skeletons: Modeling Skeletal Structures with Piecewise Algebraic Surfaces. In *Curves and Surfaces in Computer Vision and Graphics II*, pages 230–237, Boston, MA, 1991.

[5] Bajaj, C., and Ihm, I. Algebraic surface design with Hermite interpolation. *ACM Transactions on Graphics*, 19(1):61–91, January 1992.

[6] Beeker, E. Smoothing of Shapes Designed with Free Form Surfaces. *Computer Aided Design*, 18(4):224–232, 1986.

[7] Chiyokura, H., and Kimura, F. Design of Solids with Free-form Surfaces. Computer Graphics, 17(3):289–298, 1983.

[8] Chui, C. *Multivariate Splines*. Regional Conference Series in Applied Mathematics, 1988.

[9] Dahmen, W. Smooth piecewise quadratic surfaces. In T. Lyche and L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design*, pages 181–193. Academic Press, Boston, 1989.

[10] Dahmen, W. and Michelli, C. Recent Progress in Multivariate Splines. In L. Schumaker C. Chui and J. Word, editors, *Approximation Theory IV*, pages 27–121. Academic Press, 1983.

[11] deBoor, C., and Hollig, K., and Sabin, M. High Accuracy Geometric Hermite Interpolation. *Computer Aided Geometric Design*, 4(00):269–278, 1987.

[12] Farin, G. Triangular Bernstein-Bézier patches. *Computer Aided Geometric Design*, 3:83–127, 1986.

[13] Garrity, T., and Warren, J. Geometric continuity. *Computer Aided Geometric Design*, 8:51–65, 1991.

[14] Golub, G., and Van Loan, C. *Matrix Computation*. The Johns Hopkins Univ. Press, Baltimore, MD, 1983.

[15] Gregory, J., and Charrot, P. A C^1 Triangular Interpolation Patch for Computer Aided Geometric Design. *Computer Graphics and Image Processing*, 13:80–87, 1980.

[16] Hagen, H., and Pottmann, H. Curvature Continuous Triangular Interpolants. *Mathematical Methods in Computer Aided Geometric Design*, pages 373–384, 1989.

[17] Herron, G. Smooth Closed Surfaces with Discrete Triangular Interpolants. *Computer Aided Geometric Design*, 2(4):297–306, 1985.

[18] Hollig, K. Multivariate Splines. SIAM J. on Numerical Analysis, 19:1013–1031, 1982.

[19] Ihm, I.,. *Surface Design with Implicit Algebraic Surfaces*. PhD thesis, Purdue University, August 1991.

[20] Lee, E. The rational Bézier representation for conics. In G. Farin, editor, *Geometric Modeling : Algorithms and New Trends*, pages 3–19. SIAM, Philadelphia, 1987.

[21] Liu, D., and Hoschek, J. GC^1 Continuity Conditions Between Adjacent Rectangular and Triangular Bezier Surface Patches. Computer Aided Design, 21:194–200, 1989.

[22] Lorensen, W., and Cline, H. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics*, 21:163–169, 1987.

[23] Moore, D., and Warren, J. Approximation of dense scattered data using algebraic surfaces. In *Proc. of the 24th Hawaii Intl. Conference on System Sciences*, pages 681–690, Kauai, Hawaii, 1991.

[24] Nielson, G. A Transfinite Visually Continuous Triangular Interpolant. In G. Farin, editor, *Geometric Modeling Applications and New Trends*. SIAM, 1986.

[25] Patrikalakis, N., and Kriezis, G. Representation of piecewise continuous algebraic surfaces in terms of B-splines. *The Visual Computer*, 5(6):360–374, Dec. 1989.

[26] Peters, J. Local Cubic and BiCubic C^1 Surface Interpolation with Linearly Varying Boundary Normal. *Computer Aided Geometric Design*, 7:499–516, 1990.

[27] Peters, J. Smooth interpolation of a mesh of curves. *Constructive Approximation*, 7:221–246, 1991.

[28] Peters, J. Parametrizing singularly to enclose data points by a smooth parametric surface. In *Proc. of Graphics Interface*, Calgary, Alberta, June 1991. Graphics Interface '91.

[29] Piper, B. Visually Smooth Interpolation with Triangular Bezier Patches. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*. SIAM, 1987.

[30] Powell, M., and Sabin, M. Piecewise Quadratic Approximations on Triangles. *ACM Trans. on Math. Software*, 3:316–325, 1977.

[31] Preparata, F., and Shamos, M. *Computational Geometry, An Introduction*. Springer Verlag, 1985.

[32] Ramshaw, L. Beziers and B-splines as Multiaffine Maps. In *Theoretical Foundations of Computer Graphics and CAD*. Springer Verlag, 1988.

[33] Sarraga, R. G^1 interpolation of generally unrestricted cubic Bézier curves. *Computer Aided Geometric Design*, 4:23–39, 1987.

[34] Sederberg, T. Piecewise Algebraic Surface Patches. *Computer Aided Geometric Design*, 2:53–59, 1985.

[35] Seidel, H-P. A New Multiaffine Approach to B-splines. *Computer Aided Geometric Design*, 6:23–32, 1989.

[36] Semple, J., and Roth, L. *Introduction to Algebraic Geometry*. Oxford University Press, Oxford, U.K., 1949.

[37] Shirman, L., and Sequin, C. Local Surface Interpolation with Bezier Patches. Computer Aided Geometric Design, 4:279–295, 1987.

[38] Walker, R. *Algebraic Curves*. Springer Verlag, New York, 1978.

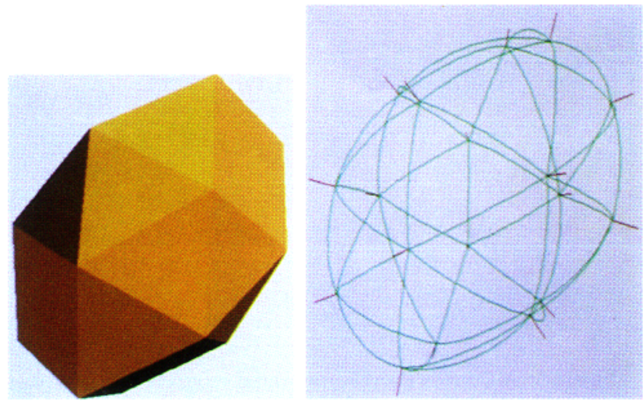

Figure 2: A Convex Polyhedron and its C^1 Conic Wireframe

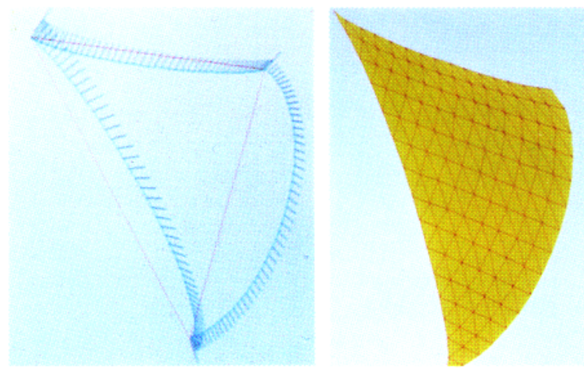

Figure 3: A Quadric Triangle and its C^1 Patch

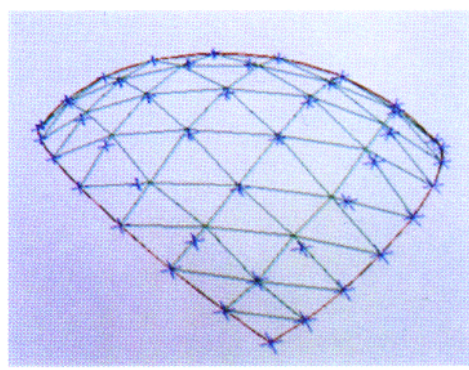

Figure 4: A Triangulation for Display and Additional (+) Points for Shape Control

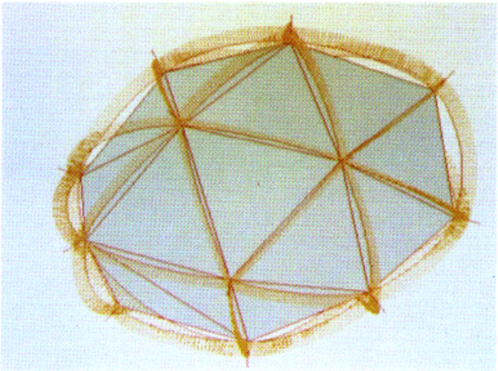

Figure 5: A Convex Polyhedron with Quadric Wires : $\rho = 0.4$

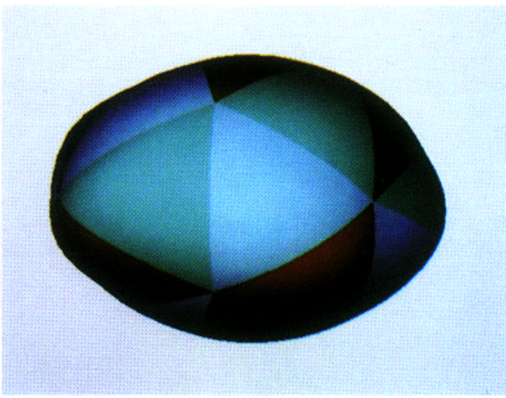

Figure 6: A C^1 Smooth Polyhedron with Quintic Algebraic Patches : $\rho = 0.4$

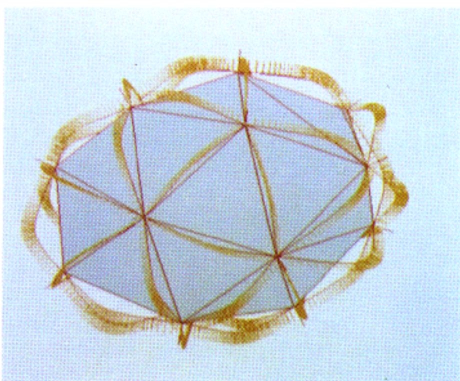

Figure 7: A Convex Polyhedron with Quadric Wires : $\rho = 0.75$

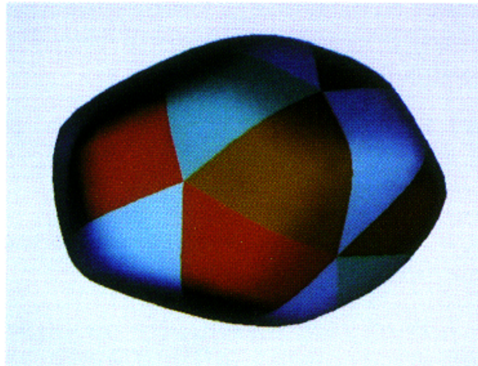

Figure 8: A C^1 Smooth Polyhedron with Quintic Algebraic Patches : $\rho = 0.75$

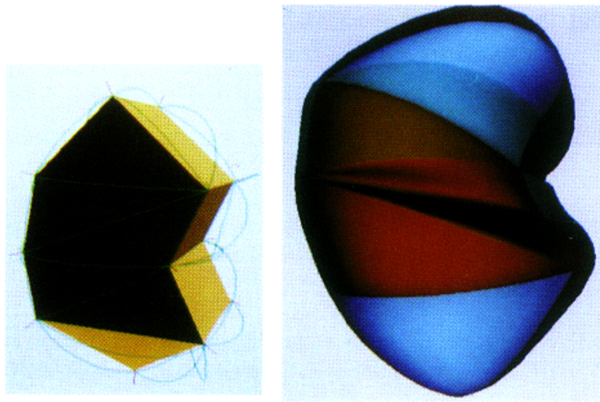

Figure 9: Smoothing the Non-Convex Polyhedron with Quintic Algebraic Patches

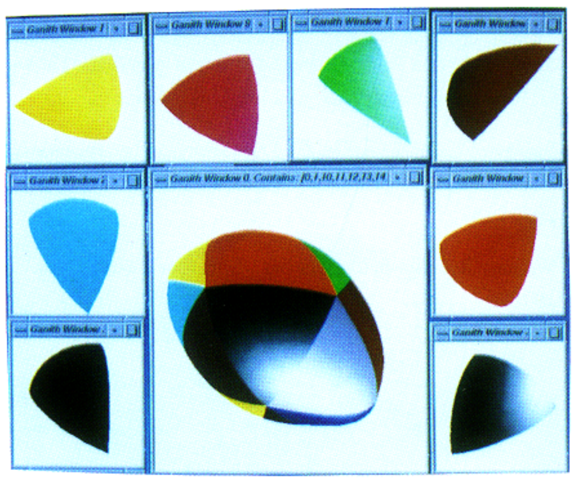

Figure 10: A Polyhedron Smoothed in the SHASTRA Distributed and Collaborative Geometric Design Environment