C1 Smoothing of Polyhedra with Implicit Surface Patches

Chanderjit L. Bajaj

Insung Ihm

Report Number:
91-060
C1 SMOOTHING OF POLYHEDRA
WITH IMPLICIT ALGEBRAIC
SURFACE PATCHES

Chanderjit L. Bajaj
Insung Ihm

CSD-TR-91-060
August 1991
Abstract

Polyhedral "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 the input polyhedron. Individual facets 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. Asymptotic degree bounds are also given for the lowest degree implicitly defined, algebraic splines required to $C^1$-smooth the vertices, edges, and facets of an arbitrary polyhedron in three dimensional real space $\mathbb{R}^3$.
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. Chui [14] and DeBoor [16] summarize much of the history of previous work. Prior work on splines have traditionally worked with a given planar triangulation using a polynomial function basis. More recently surface fitting has been considered over closed triangulation in three dimensions using a parametric surface for each triangular face [1, 9, 10, 13, 19, 20, 22, 27, 28, 36, 38].

Little work has been done on spline basis for implicitly defined algebraic surfaces. Sederberg [35] shows how various smooth implicit algebraic surfaces can be manipulated as functions in Bézier control tetrahedra with finite weights. However the problem of selecting weights for a $C^1$ mesh of implicit algebraic surface patches was left open. Dahmen [15] presents the construction of tangent plane continuous, piecewise quadric surfaces. In his construction a macro patch is split into six micro quadratic triangular patches, similar to the splitting scheme of Powell-Sabin [31]. The resulting surface patches interpolate finite sets of essentially arbitrary points in $\mathbb{R}^3$ according to a given topology (triangulation) and given normal directions at the points within some allowed ranges. The technique however works only if the original triangulation of the data set allows a transversal system of planes and hence is quite restricted. Bajaj and Ihm [7] show how blending and joining algebraic surfaces can be computed via $C^1$ interpolation. The problem of constructing a $C^1$ mesh of implicit algebraic surface patches was again left open. Moore and Warren [26] extend the marching cubes scheme of [25] and compute a $C^1$ piecewise quadratic approximation to the data within subcubes.

In this paper we consider an arbitrary spatial triangulation $T$ consisting of vertices $p = (x_i, y_i, z_i)$ in $\mathbb{R}^3$ (or more generally a simplicial polyhedron $P$ 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 $T$ or simplicial polyhedron $P$, and $C^1$ interpolates all the vertices $p_i = (x_i, y_i, z_i)$ in $\mathbb{R}^3$. Our technique is completely general and uses a single implicit surface patch for each triangular face of $T$ or $P$, i.e. no local splitting of triangular faces. Each triangular surface patch has local degrees of freedom which are used to provide local shape control. This is achieved by use of weighted least squares approximation from points $q_k = (x_k, y_k, z_k)$ generated locally for each triangular patch from the original patch data points and normal directions on them.

Why algebraic surfaces? A real algebraic surface $S$ in $\mathbb{R}^3$ is implicitly defined by a single polynomial equation $F: f(x,y,z) = 0$, where coefficients of $f$ are over the real numbers $\mathbb{R}$. Manipulating polynomials, as opposed to arbitrary analytic functions, is computationally more efficient. Furthermore algebraic surfaces provide enough generality to accurately model almost all complicated rigid objects.

Why implicit representations? While all real algebraic surfaces have an implicit definition $F$ only a small subset of these real surfaces can also be defined parametrically by the triple $G(s,t) = (x = G_1(s,t), y = G_2(s,t), z = G_3(s,t))$ 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 $F$ is its closure properties under modeling operations such as intersection, convolution, offset, blending, etc. The smaller class of parametrically defined algebraic surfaces $G(s,t)$ are not closed
under any of these operations. Closure under modeling operations allow cascading repetitions\(^1\) without any need of approximation. Furthermore, designing with a larger class of 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, most prior approaches to interpolation and least-squares scattered data fitting, have focused on the parametric representation of surfaces [17, 30, 34, 39]. Our aim here is to exhibit that implicitly defined algebraic surfaces are also very appropriate for computer aided geometric design.

The degree of an algebraic surface is the number of intersections between the surface and a line, counting complex, infinite and multiple intersections. This degree is also the same as the degree of the defining polynomial. The degree of an algebraic space curve is the number of intersections between the curve and a plane, counting complex, infinite and multiple intersections. The degree of an algebraic curve segment given as the intersection curve of two algebraic surfaces is also no larger than the product of the degrees of the two surfaces. Furthermore, the degree of a rational algebraic curve is the same as the maximum degree of the numerator and denominator polynomials in the defining triple of rational functions.

The use of low degree surface patches to construct models of physical objects results in faster computations for subsequent geometric model manipulation operations such as computer graphics display, animation, and physical object simulations. The main results of this paper are:

1. an efficient algorithm in sections 3, 4, 5 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 sections 6, 7 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.

3. a heuristic, yet effective, scheme in section 7.2 for the polygonalization and thereby display of triangular algebraic surface patches.

Both our solution surface degree bounds 5 and 7 are also significantly better than the degree 18, parametric bicubic surface patch solutions for the same problem achieved by Sarraga [34] and Peters [30]. Details on the implementation of our algorithms and illustrative examples are given in the last section.

2 \(C^1\) Continuity and Compatibility Conditions

A real algebraic space curve can be implicitly defined as the common intersection of two or more real algebraic surfaces \(C : (f_1(x, y, z) = 0, f_2(x, y, z) = 0, f_3(x, y, z) = 0, ...).\) A smaller class of rational algebraic space curves can also be represented by the triple \(\mathcal{H}(s) : (z = H_1(s), y = H_2(s), x = H_3(s)),\) where \(H_1, H_2\) and \(H_3\) are rational functions in \(s\) over \(\mathbb{R}.\) Whenever we consider the special case of a rational space curve, we assume that the curve is smooth and

\(^1\)The output of one operation acts as the input to another operation
only singly defined under the parameterization map, i.e., each triple of values for \((x, y, z)\), corresponds to a single value of \(s\).

The "normal" \(N_p\) of a point \(p\) is an arbitrary nonzero vector associated with \(p\). \(N_p\) defines a unique plane containing \(p\). The "normal" \(N_C\) of a curve \(C\) is a 1-dimensional set of vectors, one vector associated with each point \(p\) on \(C\), and orthogonal to the tangent vector at \(p\). We assume all curves are smooth i.e., nonsingular, though this is not a necessary requirement. Finally, a surface patch is defined as a smooth, connected 2-dimensional region of a surface bounded by a single cycle of curve segments.

2.1 Necessary and Sufficiency Conditions

The following definitions and lemmas are pertinent to the algorithm for the \(C^1\) smoothing of a polyhedron:

**Definition 2.1** Let \(p = (a, b, c)\) be a point with an associated "normal" \(\mathbf{m} = (m_x, m_y, m_z)\) in \(\mathbb{R}^3\). An algebraic surface \(S : f(x, y, z) = 0\) is said to contain \(p\) with \(C^1\) continuity if

1. \(f(p) = f(a, b, c) = 0\), (containment condition)
2. \(\nabla f(p)\) is not zero and \(\nabla f(p) = \alpha \mathbf{m}\), for some nonzero \(\alpha\). (tangency condition)

**Definition 2.2** Let \(C\) be an algebraic space curve with an associated varying "normal" \(\mathbf{n}(x, y, z) = (n_x(x, y, z), n_y(x, y, z), n_z(x, y, z))\), defined for all points on \(C\). An algebraic surface \(S : f(x, y, z) = 0\) is said to contain \(C\) with \(C^1\) continuity if

1. \(f(p) = 0\) for all points \(p\) of \(C\). (containment condition)
2. \(\nabla f(p)\) is not identically zero and \(\nabla f(p) = \alpha \mathbf{n}(p)\), for some \(\alpha\) and for all points \(p\) of \(C\). (tangency condition)

**Lemma 2.1** A necessary condition for smoothing a polyhedron with tangent-plane-continuous triangular surface patches is a single tangent plane at each vertex of the polyhedron.

2.2 Compatibility and Non-Singularity Constraints

We need a few basic concepts from differential geometry [12, 29]. A surface \(S \subset \mathbb{R}^3\) is regular at a point \(p \in S\) if there exists a neighborhood \(V \subset \mathbb{R}^3\) and a map \(x : U \rightarrow V \cap S\) of an open set \(U\) in \(\mathbb{R}^2\) onto \(V \cap S \subset \mathbb{R}^3\) such that \(x(u, v) = (x(u, v), y(u, v), z(u, v))\) is differentiable, homeomorphic, and its differential \(dx_q : \mathbb{R}^2 \rightarrow \mathbb{R}^3\) is one-to-one for each \(q \in U\). A surface \(S\) is regular if, at each point on \(S\), \(S\) is regular. A tangent vector to a regular surface \(S\) at a point \(p \in S\) is the tangent vector \(\alpha'(0)\) of a differentiable curve \(\alpha : (-\epsilon, \epsilon) \rightarrow S\) with \(\alpha(0) = p\). The plane \(T_p(S)\) spanned by all tangent vectors to \(S\) at \(p\), is called the tangent plane to \(S\) at \(p\) that is, in fact, a two dimensional vector space. For a regular point \(p \in S\), a unit vector which is perpendicular to \(T_p(S)\) is called a unit normal vector at \(p\). For each \(q \in x(U)\), we define a differentiable field of unit normal vectors \(N : x(U) \rightarrow \mathbb{R}^3\) such that \(N(q) = \frac{x_u(q) \times x_v(q)}{\|x_u(q) \times x_v(q)\|}(q)\), where \(x_u = \frac{\partial x}{\partial u}\) and \(x_v = \frac{\partial x}{\partial v}\). The map \(N : S \rightarrow G\), taking its values in the unit sphere, is called the Gauss map of \(S\), where \(G\) is a unit sphere. Then the Gauss map is differentiable, and its differential \(dN_p\) of \(N\) at \(p\) is a linear map from \(T_p(S)\) to \(T_p(S)\). It measures the rate of the normal vector \(N\) in a neighborhood of \(p\).
The following lemma provides a condition which must be satisfied when the unit normal vectors of a surface \( S \) change in the neighborhood of regular points. Its proof is found in Chapter 3, pp. 140 [12].

**Lemma 2.2** The differential \( dN_p : T_p(S) \rightarrow T_p(S) \) of the Gauss map is a self-adjoint linear map, that is, \( (dN_p(w_1), w_2) = (w_1, dN_p(w_2)) \) where \( w_1 \) and \( w_2 \) are two independent tangent vectors at a regular point \( p \), and \((\cdot, \cdot)\) is an inner product of two vectors.

The symmetry of the linear map \( dN_p \), implied by Lemma 2.2, entails a necessary condition that must be satisfied between tangent vectors and the rates of changes of normal vectors at a regular point. It implies that, given two regular curves passing through a regular point on a surface, the unit normal vector must change along each curve satisfying the equality in the lemma.

Consider the problem of tangent-plane-continuous interpolation of two parametric space curves with normal directions, meeting at a point. Let \( C_1(u) \) and \( C_2(v) \) be two parametric curves with parametrically specified normal directions \( N_1(u) \) and \( N_2(v) \) such that \( C_1(0) = C_2(0) = p \), and \( N_1(0) \) and \( N_2(0) \) are proportional, that is, the two curves meet at \( p \) and they share the same normal direction at the point. We look for a surface \( S \) which smoothly interpolates the curves, that is,

- \( S \) must contain \( C_1(u) \) and \( C_2(v) \),
- the normals of tangent planes of \( S \) along the curves must coincide with the normals of the curves, and
- \( S \) is regular at \( p \).

Suppose that there exist such a surface \( S \). Then, we have a local parametrization \( x : U \rightarrow V \cap S \) of an open set \( U \) in \( \mathbb{R}^2 \) onto \( V \cap S \subset \mathbb{R}^3 \) for a neighborhood \( V \) of \( p \) such that

- \( x(0,0) = p \),
- \( x_u = \frac{\partial x}{\partial u}(0,0) = C_1'(0) \) and \( x_v = \frac{\partial x}{\partial v}(0,0) = C_2'(0) \), and
- the Gauss map \( N \) of \( S \) is such that \( N(C_1(u)) = \frac{N_1(u)}{||N_1(u)||} \) and \( N(C_2(v)) = \frac{N_2(v)}{||N_2(v)||} \).

Then, by Lemma 2.2, in order for \( S \) to be regular at \( p \), it should be that

\[
(dN_p(x_u), x_v) = (x_u, dN_p(x_u)).
\] (1)

By the definition of the differential,

\[
dN_p(x_u) = \frac{dN(C_1(u))}{du} \bigg|_{u=0} = \frac{d\left(\frac{N_1(u)}{||N_1(u)||}\right)}{du} \bigg|_{u=0} = \frac{N_1'(u) || N_1(u) || - N_1(u) || N_1'(u) ||}{|| N_1(u) ||^2} \bigg|_{u=0} = \frac{N_1'(0) || N_1(0) || - N_1(0) || N_1'(0) ||}{|| N_1(0) ||^2}.
\]
3 POLYHEDRON SMOOTHING ALGORITHM

Since \((N_1(0), x_v) = 0, (dN_p(x_u), x_v) = \frac{(N_1'(0), x_u)}{\|N_1(0)\|} = \frac{(N_1'(0), C_1'(0))}{\|N_1(0)\|}\). In the same way, we get \((x_u, dN_p(x_v)) = \frac{(C_1'(0), N_2'(0))}{\|N_2(0)\|}\). Hence, the equation (1) becomes

\[
\frac{(N_1'(0), C_1'(0))}{\|N_1(0)\|} = \frac{(C_1'(0), N_2'(0))}{\|N_2(0)\|}.
\]

The above argument implies that enforcing two curves to have the same normal vectors at a common point does not guarantee the regularity of an interpolating surface at the point. The equation (2) is a necessary condition for regularity, indicating that, if the given curves and their normals do not satisfy the equation (2), any smoothly interpolating surface must be singular at \(p\).

**Theorem 2.1** 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 \(\frac{(N_1'(0), C_1'(0))}{\|N_1(0)\|} = \frac{(C_1'(0), N_2'(0))}{\|N_2(0)\|}\).

In conclusion, Lemma 2.1 and Theorem 2.1 impose necessary and sufficient conditions respectively, for the \(C^1\) smoothing of a polyhedron.

3 Polyhedron Smoothing Algorithm

We present below a sketch of the algorithm to \(C^1\) smooth a simple polyhedron \(P\) with tangent-plane-continuous implicit surface patches.

**Algorithm**

1. Triangulate each of the non-triangular polygonal faces of the given polyhedron \(P\). Any simple polygon is easily triangulable by adding non-intersecting inner diagonals[32].

2. Specify a unique “normal” vector at each vertex of \(P\). This provides a unique tangent plane for all patches which shall \(C^1\) interpolate that vertex.

3. Next, construct a curvilinear wire frame by replacing each edge of \(P\) 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.

4. Specify normal vectors at each point along each of the edge curves. This provides the tangent planes for the two incident patches which shall \(C^1\) interpolate the edge curves. If it is required that the individual patches are non-singular at the vertices of \(P\), then the variation of normals along different edge curves incident at the same vertex need also to be made compatible.

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 chosen via weighted least squares to achieve a suitably shaped single-sheeted surface patch. The resulting surface patches yield a globally \(C^1\) smooth curved model for the given polyhedron.
Details of each of the steps 2 to 5 of the algorithm for specific classes of polyhedra (convex, non-convex) as well as the explicit degrees of the required curves and surfaces are presented in subsequent sections 4, 5 and 6.

4 Wireframe Construction

4.1 Choice of Vertex Normals

The unique "normal" vector assigned to each vertex of the triangulated polyhedron \( P \) can be chosen independently and quite arbitrarily. However, the relative directions of each adjacent vertex normal pair affects the degree of the \( C^1 \) interpolating edge curve which replaces the straight edges of \( P \). Let the two normal vectors at the two endpoints of an edge be called an edge-normal-pair. Certain relative directions of an edge-normal-pair induce an inflection point (zero curvature 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 \( P \). Here we restrict ourselves to surface fitting without the splitting of any triangular faces of \( P \).

We first derive a necessary and sufficient condition for the relative directions of an edge-normal-pair to allow a \( C^1 \) conic interpolation. 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 4.1** 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 conic curve \( f(x,y) = ax^2 + 2hxy + by^2 + 2gx + 2fy + c \) such that

- \( S(P_0, P_1) \) is a continuous segment of \( f(x,y) = 0 \),
- \( p_0 \) and \( p_1 \) are the end points of \( S(P_0, P_1) \), and
- the gradients of \( f(x,y) = 0 \) at \( p_0 \) and \( p_1 \) have the same directions as \( n_0 \) and \( n_1 \), respectively. In other words, \( \frac{(\nabla f(p_0),n_0)}{\|\nabla f(p_0)\|\|n_0\|} = 1 \) and \( \frac{(\nabla f(p_1),n_1)}{\|\nabla f(p_1)\|\|n_1\|} = 1 \).

Given a pair \( P = ((p_x,p_y),(n_x,n_y)) \), we can define \( T_P(x,y) = n_x(x-p_x) + n_y(y-p_y) = 0 \) which is the equation of the tangent line that passes through \( (p_x,p_y) \) and has a normal direction \( (n_x,n_y) \). Note that the tangent line \( T_P(x,y) = 0 \) contain the same direction as \( (n_x,n_y) \), and divides a plane into a positive halfspace \( \{(x,y) \in \mathbb{R}^2 | T_P(x,y) > 0 \} \), and a negative halfspace \( \{(x,y) \in \mathbb{R}^2 | T_P(x,y) < 0 \} \).

**Lemma 4.1** Let \( p_0 \) and \( p_1 \) be on a proper conic \( f(x,y) = ax^2 + 2hxy + by^2 + 2gx + 2fy + c = 0 \). Then, \( T_{(p_0,\nabla f(p_0))}(p_1) \cdot T_{(p_1,\nabla f(p_1))}(p_0) > 0 \).

**Proof**: Without loss of generality, we assume that \( p_0 = (0,0) \), and \( p_1 = (1,0) \). Since \( \nabla f(x,y) = (2ax + 2h + 2g, 2hx + 2by + 2f) \), \( \nabla f(0,0) = (2g, 2f) \) and \( \nabla f(1,0) = (2a + 2g, 2h + 2f) \). Hence, \( T_{(p_0,\nabla f(p_0))}(x,y) = 2gx + 2fy \), and \( T_{(p_1,\nabla f(p_1))}(x,y) = (2a + 2g)(x-1) + (2h + 2f)y \). From the containment conditions of the two points, \( f(0,0) = c = 0 \), and \( f(1,0) = a + 2g + c = 0 \). Then,
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 \( T_{p_0} \). The following theorem shows that this condition is, in fact, a sufficient and necessary condition.

**Theorem 4.1** There exists a conic segment \( S(p_0, p_1) \) that smoothly interpolates two pairs \( p_0 = (p_0, n_0) \) 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 \( f(x, y) = 0 \) be a conic for a smoothly containing conic segment. From our definition of smooth interpolation, it follows that \( T_{p_0}(p_1) \cdot T_{p_1}(p_0) = T(p_0, \nabla f(p_0))(p_1) \cdot T(p_1, \nabla f(p_1))(p_0) \) which is positive according to Lemma 4.1.

\( \Leftarrow \) If \( T_{p_0}(p_1) \cdot T_{p_1}(p_0) > 0 \), then the conic in \( q(x, y) = L(x, y)^2 - \kappa \cdot T_{p_0}(x, y) \cdot T_{p_1}(x, y) = 0 \) or \( -q(x, y) = 0 \) will smoothly interpolate the two pairs where \( L(x, y) = 0 \) is the line connecting \( p_0 \) and \( p_1 \), and \( \kappa \) is a constant [33].

Now, back to the original problem of computing a quadric wire smoothly interpolating two given point and unit normal vector pairs \( p_0 = (p_0, n_0) \) and \( p_1 = (p_1, n_1) \) in \( \mathbb{R}^3 \). The concept of the tangent line in a plane is naturally extended to an oriented tangent plane \( T_p(x, y, z) = n_x(x - p_x) + n_y(y - p_y) + n_z(z - p_z) = 0 \) given \( P = ((p_x, p_y, p_z), (n_x, n_y, n_z)) \) in 3D space, and this tangent plane divides 3D space into two halfspaces. In fact, we see that the inequality \( T_{p_0}(p_1) \cdot T_{p_1}(p_0) > 0 \) is also a criterion which determines if a quadric wire can smoothly interpolate two given pairs of points and normal vectors.

**Corollary 4.1** Given two point and unit normal vector pairs \( p_0 = (p_0, n_0) \) and \( p_1 = (p_1, n_1) \) in 3D space, there exists a quadric wire \( W(t) = (C(t), N(t)) \), contained in a plane determined by a given plane normal vector \( npl_{01} \), that smoothly interpolates the pairs if and only if \( T_{p_0}(p_1) \cdot T_{p_1}(p_0) > 0 \).

**Proof**: Consider the two pairs \( p_0 \) and \( p_1 \), their two tangent planes \( T_{p_0} \) and \( T_{p_1} \), and the plane \( H \) which is defined by \( npl_{ij} \). Then, the intersection lines of \( H \) and \( T_{p_0} \) and \( T_{p_1} \) become the tangent lines in \( H \) to which a conic curve must be tangent. That is, the normal vectors of the tangent lines are the projections of the normal vectors of the tangent planes. Note that the positiveness and negativeness of halfspaces are inherited from 3D space to the plane \( H \). Hence, we see that the inequality \( T_{p_0}(p_1) \cdot T_{p_1}(p_0) > 0 \) holds in 3D space if and only if its 2D version holds in \( H \).

If there exists a conic curve in \( H \), we can find a quadric surface which smoothly interpolates the given pairs, as explained before, and take \( W(t) \) from this quadric surface that has the same gradient directions as the given two normal vectors. 

4.2 Generation of a Conic Wireframe

First, we give a definition of the quadric wire.

---

Thanks to J. Yu for suggesting this reference.
Definition 4.2 Let $C(t) = (x(t), y(t), z(t))$ and $N(t) = (nx(t), ny(t), nz(t))$ 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 quadratic surface $q(x, y, z) = 0$ such that $q(C(t)) = 0$ and $\nabla q(C(t))$ is proportional to $N(t)$ for all real $t$.

The first step to smoothing a convex polyhedron is to compute a conic curve given two point and unit normal vector pairs $(p_0, n_0), (p_1, n_1)$ and a normal $n_{pl}$ of a plane such that

1. the computed conic curve passes through $p_0$ and $p_1$,
2. its tangents at $p_0$ and $p_1$ are perpendicular to $n_0$ and $n_1$, respectively, and
3. it is contained in the plane which contains $p_0$ and $p_1$, and has the plane normal $n_{pl}$.

Especially, we force $W(0) \equiv (p_0, n_1)$ and $W(1) \equiv (p_1, n_1)$. Then we use a 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 be. (See Figure 1). This projection results in a control triangle $p_0 - p_2 - p_1$. Lee [24] 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 given by the special choice $w_0 = w_1 = 1 - \rho, w_2 = \rho, \rho > 0$. By introducing the parameter $\rho$, we can control the shape of a conic intuitively. Let $p_{01} = (p_0 + p_1)/2$ be the midpoint of the chord $p_0 p_1$. Then, $\rho$ has a property that $C(0.5) - p_{01} = \rho (p_2 - p_{01})$. From this, we can see that as $\rho$ is increased, the conic gets more curved. In particular, it can be proven that $\rho = 0.5$ for parabola, $0 < \rho < 0.5$ for ellipse and $0.5 < \rho < 1.0$ for hyperbola.

$^3$By $\equiv$, we mean the points are the same, and the normal vectors are proportional.
4.3 Assigning Normals along Edge curves

Once \( C(t) \) is fixed, we find a quadratic surface \( q(x, y, z) = 0 \) such that \( N(t) \), which is a restriction of \( \nabla q(x, y, z) \), interpolates \( n_0 \) and \( n_1 \). Consider a quadratic surface \( q(x, y, z) = c_0 x^2 + c_1 y^2 + c_2 z^2 + c_3 x y + c_4 x z + c_5 y z + c_6 x + c_7 y + c_8 z + c_9 = 0 \). \( q(x, y, z) = 0 \) 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 \( C(t) \). Our Hermite interpolation algorithm gives 5 linear equations in terms of the unknowns \( c_i \) for the containment requirement. It is obvious that 5 constraints on \( c_i \) are required considering the Bezout theorem which says if a conic intersects with a quadratic surface at more than 4 points, the curve is contained in the surface.

Hence, \( 4 (= 9 - 5) \) degrees of freedom in choosing \( c_i \) are left, and these must be 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 quadratic surface. But we can see that requiring only one more normal vector at a point on the curve fixes the normal vectors along the whole conic. Consider the gradient vector \( \nabla q(x, y, z) \) whose components are linear. Then, the vector function \( \nabla q(C(t)) \) is a degree 2 polynomial parametric curve in the projective space, and hence, three independent constraints fixes the curve \( \nabla q(C(t)) \), or the normal vector along \( C(t) \). After we specify one more normal vector at a point on the conic, we obtain a family of quadratic surfaces \( q(x, y, z) \) with one degree of freedom where all the surfaces in the family contain \( C(t) \), and share the same gradient vectors along \( C(t) \). This observation leads to the following lemma:

**Lemma 4.2** Let \( W(t) = (C(t), N(t)) \) be a quadric wire. Then, the quadratic surfaces which smoothly interpolate \( W(t) \) comprises a family of surfaces with one degree of freedom.

What we do in our implementation in order to fix the normal vector is the following: first, the average \( n_{01} = (n_0 + n_1) / 2 \) is computed, and then \( n_{01} \) is projected into a plane which contain \( C(0.5) \), and is perpendicular to the tangent vector \( C'(0.5) \). Then, we require the projected vector to be the normal vector at \( C(0.5) \). Once the normal vectors along \( C(t) \) is fixed, we define \( N(t) \) to be the vectors.

4.4 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 is what has been used in the past and previously reported for example in [19, 20, 34] We therefore omit further discussion of this construction and refer the reader to the earlier references.

5 Local Interpolatory Patch Generation

**Definition 5.1** An augmented triangle is an 9-tuple \( T = (p_0, p_1, p_2, n_0, n_1, n_2, n_{p01}, n_{p12}, n_{p20}) \) where the points \( p_i \) are three vertices of a triangle with the corresponding unit normal vectors
5 LOCAL INTERPOLATORY PATCH GENERATION

Figure 2: A Triangular Curved Wireframe and the $C^1$ Surface Patch

$n_i$, and $n_{pi,j}$ is the normal of the plane which will contain the quadric wire made from $(p_i, n_i)$ and $(p_j, n_j)$.

**Definition 5.2** 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 subsection. Now the quadric triangle is to be fleshed using an algebraic surface $f(x, y, z) = 0$. The algebraic surface to be used should be flexible enough to interpolate the three quadric wires smoothly, i.e., with tangent plane continuity, see Figure 2. Though higher degree algebraic surfaces provide more flexibility, the number $(n^3)$ of coefficients of a degree $n$ algebraic surface grows dramatically as $n$ increases. Hence, for fast computation and less numerical errors, keeping the degree of a surface in a reasonable range is very important.

We first compute general degree bounds for interpolatory triangular patches with degree $d$ 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 smoothly interpolate a wire of degree $d W(t) = (C(t), N(t))$. According to the Bezout theorem, $dn + 1$ constraints on the coefficients of $f$ are required for $f$ to contain $C(t)$ which is of degree $d$. For tangent plane 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 the degree $d(n - 1)$. This vector function is, in fact, a degree $d(n - 1)$ parametric polynomial curve in the projective space. Hence $d(n - 1) + 1$ independent constraints are enough to fix the gradient of $f$ along the curve $C(t)$, making $\nabla f(C(t))$ proportional to $N(t)$ which is the requirement of tangent plane continuity. This yields the following lemma.
Lemma 5.1 Let $W(t) = (C(t), N(t))$ be a degree $d$ wire. For an algebraic surface $f(x, y, z) = 0$ of degree $n$ to smoothly interpolate $W(t)$, at most $2dn - d + 2 = d(n + 1 + d(n - 1) + 1)$ independent linear constraints on the $f$'s coefficients must be satisfied.

For $C^1$ interpolation of a triangular patch there exists a geometric dependency between the three wires which also leads to dependency amongst the $C^1$ constraints. First, since the curves intersect pairwise, there must be three rank deficiencies between the equations from the containment conditions. 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, for each curve, there are two rank deficiencies between the linear equations for the containment conditions, and the equations for its tangency condition. Hence, 6 additional rank deficiencies with the previous 3 yield a total of 9 overall deficiencies.

Lemma 5.2 Let $QT = (W_0(t), W_1(t), W_2(t))$ be a conic triangle. The rank of the linear system $M_{1X} = 0$ which is constructed by the Hermite interpolation algorithm for the algebraic surface $f(x, y, z) = 0$ of degree $n$ that smoothly fleshes $QT$, is at most $12n - 9$.

Proof: For $C^1$ of all three conic wires requires $3(4n - 2 + 2) = 12n$. Using lemma 5.1 minus the 9 deficiencies as shown above, yields the bound.

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 5.3 Let $QT = (W_0(t), W_1(t), W_2(t))$ be a cubic triangle. The rank of the linear system $M_{1X} = 0$ which is constructed by the Hermite interpolation algorithm for the algebraic surface $f(x, y, z) = 0$ of degree $n$ that smoothly fleshes $QT$, is at most $18n - 12$.

Proof: For $C^1$ of all three cubic wires requires $3(6n - 3 + 2) = 18n - 3$ using lemma 5.1 minus the 9 deficiencies.

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.

\*Again, for each curve, we can choose point-normal pairs at the two end points. The resulting two linear equations are linearly dependent on the equations from the containment requirement.
Lemma 5.4 Let $QT = (W_0(t), W_1(t), W_2(t))$ be a conic triangle with one edge a cubic curve. The rank of the linear system $M_1 x = 0$ which is constructed by the Hermite interpolation algorithm for the algebraic surface $f(x, y, z) = 0$ of degree $n$ that smoothly fleshes $QT$, is at most $14n - 10$.

Proof: For $C^1$ of two conics and a cubic wire requires $2(4n - 2 + 2) + (6n - 3 + 2) = 14n - 1$ using lemma 5.1 minus the 9 deficiencies.

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 5.5 Let $QT = (W_0(t), W_1(t), W_2(t))$ be a cubic triangle with one edge a conic curve. The rank of the linear system $M_1 x = 0$ which is constructed by the Hermite interpolation algorithm for the algebraic surface $f(x, y, z) = 0$ of degree $n$ that smoothly fleshes $QT$, is at most $16n - 11$.

Proof: For $C^1$ of two cubics and a conic wire requires $(4n - 2 + 2) + 2(6n - 3 + 2) = 16n - 2$ using lemma 5.1 minus the 9 deficiencies.

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.

6 Surface Selection and Local Shape Control

As a result of smooth interpolation of a quadric triangle $QT$ with a degree 5 surface, a family of algebraic surfaces $f(x, y, z) = 0$ with, at least, 4 degrees of freedom is obtained. Similarly $C^1$ interpolation of a cubic triangle is achieved with a 5 parameter family of degree 7 surfaces. The family is expressed as the nontrivial coefficients vectors in the nullspace of $M_1$. To select a degree 5 or 7 surface from the respective families, those 4 degrees of freedom must be specified. We show that least squares approximation to additional points around the triangular patch, is well suited for this purpose.

Let $S_0 = \{v_i \in \mathbb{R}^3 | i = 1, \cdots, l\}$ be a set of points which approximately describes a desirable surface patch. Then, we can get a linear system $M_A x = 0$, where each row of $M_A$ is obtained from $f(v_i) = 0$. Then the conventional least squares approximation is to minimize $\|M_A x\|^2$ over the nullspace of $M_1$. However, our experiments show that in many cases, singularities occur inside the quadric triangle. Just minimizing $\|M_A x\|^2$ makes the resulting surface well approximate the set of points, however, this simple algebraic approximation can not prevent the resulting surface from self-intersecting inside the triangle.

To rid our solution surfaces of 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. Consider some smooth region of an algebraic surface. Since the derivatives of $w = f(x, y, z)$ are well defined in the region, the contour levels behaves well in the proximity of the zero contour. In our scheme, we first generate $S_0 =$
7 Computational Details

7.1 Solution of Interpolation and Least-Squares Matrices

The $C^1$ interpolation algorithm takes as input positional and first derivative (normal) information on points and algebraic space curves. For an algebraic surface $S: f(x,y,z) = 0$ of degree $n$, it produces a homogeneous linear system $M_I x = 0$, $M_I \in \mathbb{R}^{n_x \times n_U}$ of $n_I$ equations and $n_U$ unknowns where $x$ is a vector of the $n_U (= \binom{n+3}{3})^5$ coefficients of $S$.

Then, the nontrivial solutions in the nullspace of $M_I$ 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 $q = n_U - r$ 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, $n_U - r - 1$ degrees of freedom in choosing an instance surface from the family. Hence, the rank $r$ of $M_I$ must be less than the number of the coefficients $n_U$, should there exist an interpolating surface.

A matrix $M_A \in \mathbb{R}^{n_A \times 56}$ for least-squares approximation is next constructed, similar to the construction of $M_I$, for the additional points generated around the triangular patch as described in section 6. 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.

$$
\begin{align*}
\text{minimize} & \quad \| M_A x - b \|_2 \\
\text{subject to} & \quad M_I x = 0,
\end{align*}
$$

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

To find the nullspace of $M_I$ in a computationally stable manner, the singular value decomposition (SVD) of $M_I$ is computed [18] where $M_I$ is decomposed as $M_I = U \Sigma V^T$ where $U \in \mathbb{R}^{n_I \times n_U}$ and $V \in \mathbb{R}^{56 \times 56}$ are orthonormal matrices, and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_s) \in \mathbb{R}^{n_I \times 56}$ is a diagonal matrix with diagonal elements $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_s \geq 0$ ($s = \min\{n_U, 56\}$). It is known that the rank $r$ of $M_I$ is the number of the positive diagonal elements of $\Sigma$, and that the last $56 - r$ columns of $V$ span the nullspace of $M_I$. Hence, the nullspace of $M_I$ is expressed as:

$\binom{n+3}{3}$ There are $\binom{n+3}{3}$ coefficients in $f(x,y,z)$ of degree $n$. 

$$\{(v_i, n_i) | i = 1, \ldots, l\}$$ 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 $S_1 = \{v_i | v_i = v_i + \alpha n_i, i = 1, \ldots, l\}$, and $S_{-1} = \{w_i | w_i = v_i - \alpha n_i, i = 1, \ldots, l\}$ for some small $\alpha > 0$. Then, we get the least squares system $M_A = b$ from three kinds of equations: $f(v_i) = 0$, $f(w_i) = 1$, 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 a well behaved contour levels gets rid of self-intersection in the region. We give an algorithm for generation of the point-normal set $S_0$ in the last paragraph of Subsection 7.2.
\{ x \in \mathbb{R}^{56} \mid x = \sum_{i=1}^{56-r} w_i v_{r+i}, \text{ where } w_i \in \mathbb{R}, \text{ and } v_j \text{ is the } j\text{th column of } V \}, \text{ or } x = V_{56-r} w \text{ where } V_{56-r} \in \mathbb{R}^{56 \times (56-r)} \text{ is made of the last } 56-r \text{ columns of } V, \text{ and } w \text{ a } (56-r)\text{-vector.}^6 \quad x = V_{56-r} w \text{ compactly expresses all the quintic surfaces which Hermite-interpolate the three quadric wires.}

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

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

where \( R_1 \in \mathbb{R}^{(56-r) \times (56-r)} \) is upper triangular. (This factorization is called a \( Q-R \) factorization [18]). Now, let

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

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

### 7.2 Display of the Triangular Algebraic Patch

As implicitly defined algebraic surfaces have become increasingly important in geometric modeling, several algorithms for displaying them have emerged. Implicit algebraic surfaces lend themselves naturally to ray tracing [21]. Sederberg and Zundel [37] use a scan line display method which offers improvement in speed and correctly displays singularities. Even though both approaches produce images of good qualities, the computational cost is high. Also, the static processes do not allow interactive display of surfaces. On the other hand, polygonization of implicit surfaces [2, 11] can use the capability of the graphics hardware which provides very fast interactive rendering. Allgower [2] uses simplices to approximate a surface with polygonal meshes. In [11], Bloomenthal presents a numerical technique that approximates an implicit surface with a polygonal representation. The technique is to surround the implicit surface with an octree, at whose corners the implicit function is sampled to generate polygons. Both these techniques work well for the entire algebraic surface but are not well suited to our purpose of polygonalizing an implicit triangular surface patches. The major problem here is to isolate only the necessary portion of the triangular patch from the entire implicit surface.

In our display routine, we \textit{March} via space curve tracing [6], over the necessary triangular patch regions producing polygons which approximate the surface patches.

The following simple recursive procedure produces adaptive polygonization of a triangular algebraic surface patch. Let \( f(x, y, z) = 0 \) be a primary surface whose triangular portion clipped by three planes \( h_i(x, y, z) = 0, \ i = 1, 2, 3 \) is to be polygonized. (See Figure 3.) Initially, the triangle \( T_0 = (P_0, P_1, P_2) \) is a rough approximation of the surface patch. Each boundary curve decided by \( f \) and \( h_i \) is traced producing a digitized linear approximation to the space curves, then the linear approximation is segmented adaptively into a new segment of order \( 2^a \) for some

---

6 As mentioned before, in most cases, the rank \( r \) of \( M_f \) 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.
given $d$. (See, for example, [23] for an adaptive segmentation algorithm of space curves.) Then, $T_0$ is refined into four triangles by introducing the 3 points $Q_0$, $Q_1$, and $Q_2$ where $Q_i$, $i = 0, 1, 2$ is the center point of each adaptive segmentation of order $2^d$. The clipping planes of subdivided triangles can be computed by averaging the normals of the two triangles incident to the edge. Then, each new edge is traced, and then its adaptive linear approximation of order $2^d-1$ is produced. In this way, this new approximation is further refined by recursively subdividing each triangle until some stopping criterion is met.

While the method produces a regular, but adaptive, network of polygons, it might be improved to generate a more adaptive polygonization. Rather than subdividing all the triangles up to the same level, each triangle is examined to see if it is already a good approximation to the surface portion it is approximating. It is refined only when the answer is no. Criteria for such local refinement are suggested in [2, 11]. However, to design an irregular adaptive polygonization algorithm with robust local refinement criterions, is an open problem.

We also use the above recursive subdivision scheme to produce $S_0 = \{(w_i, n_i)|i = 1, \ldots, l\}$ in Subsection 7.1. Initially, only the boundary curves are known, and each time a new curve is to be traced in the algorithm, a quadric wire is computed as explained in Subsection 4 from the information on the initial and final points, their normals and clipping plane. The generated quadric wire gives approximate curve and normal information, and is traced to generate points and normals. The final polygonal approximation obtained in this way gives a set of points which is used in least squares approximation. We observe that this heuristic method work quite well when the $\rho$ value is in the reasonable range, say, $0.25 \leq \rho \leq 0.75$. Figure 4 displays
a polygonization of a triangular algebraic surface patch, and the points used for shape control via weighted least squares approximation, discussed in section 6.

8 Remarks and Open Problems

8.1 Implementation Issues

We have presented a method that smooths out a polyhedron with tangent-plane-continuous piecewise implicitly represented triangular algebraic surface patches. The polyhedron smoothing algorithms have been implemented in our geometric modeling toolkit SHILP [3] in conjunction with our algebraic geometry toolkit GANITH [8]. For polyhedron smoothing, SHILP takes as input a polyhedron $P$ and a user specified $\rho$ value (for shape control), and computes a combination of quadric wires (if the normal condition is satisfied for the edge) and cubic wires together with the variation of normals along the curves. Next, for each triangular facet of curves GANITH is invoked via inter process communication and the facet $C^1$ fitted with a low degree (5 to 7) algebraic surface patch. Then polygonized triangular patches are rendered interactively in a display window of SHILP.

8.1.1 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 $T = (p_0, p_1, p_2, n_0, n_1, n_2, n_{pl01}, n_{pl12}, n_{pl20})$ 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 $n_i$ at each vertex $p_i$ can be computed by averaging the normals of the faces incident to the vertex. Other assignment schemes which rely on the normals arising from a sphere or a paraboloid are also possible. For a convex triangulation $T$ or polyhedron $\mathbf{P}$, the above choice of normals at vertices always yields compatible vertex-normal pairs (as per section 4) for $C^1$ conic interpolation and hence degree five surface patches suffice by results in section 5. 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 which we discuss a little later in this section. 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 5) Prove here that the averaging of plane normals at vertices also 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 $n_{plij}$. After the normal data is computed, quadric wires are generated for the $\rho$ value which is interactively controlled by the user.

Example 8.1 Construction of Quadric Wire Frames
Figure 5: A Convex Polyhedron with Quadric Wires: $\rho = 0.4$

Figure 5 and 6 show two quadric wire frames for the same convex polyhedron with the $\rho$ values 0.4 and 0.75, respectively. □

**Example 8.2 Polyhedrons Smoothed by Using Quintic Implicit Algebraic Surfaces**

Each of 32 faces of the polyhedron in Example 8.1 is replaced by a quintic implicit algebraic surface which smoothly fleshes its quadric triangle. The result is the piecewise tangent-plane-continuous quintic algebraic surface meshes which smooth the given polyhedron. Figure 7, and 8 respectively illustrate the $C^1$ surface meshes of $\rho = 0.4$ and 0.75. As explained before, ellipses, and hyperbolas are used as quadric wires for $\rho = 0.4$ and 0.75, respectively. □

**8.2 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 least squares approximation. While the heuristics for weighted least square approximation usually work very well, sometimes we need to manually change, for example, the value of $\alpha$ in $S_1$ and $S_2$ of Subsection 7.1. 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 quintic surface patch that smoothly fleshes a nonconvex combination of quadric wires.) An open problem is to construct a wire frame for a nonconvex 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. An 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 quintic algebraic surface patches, and to manipulate them through geometric operations such as boolean set operations.

---

7This polyhedron is gyroelongated triangular bicupola with its rectangular faces triangulated.
Figure 6: A Convex Polyhedron with Quadric Wires: $\rho = 0.75$

Figure 7: $C^1$ Smooth Polyhedron with Quintic Algebraic Surfaces: $\rho = 0.4$
Figure 8: $C^1$ Smooth Polyhedron with Quintic Algebraic Surfaces: $\rho = 0.75$

Figure 9: Smoothing a Nonconvex Polyhedron with Quintic Algebraic Surface Patches
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 SHILP and GANITH with the VAIDAK toolkit [4]. See Figure 10. For algorithmic details of the skeletal model reconstruction see [5].

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

References


REFERENCES


REFERENCES


