1994

Smooth Low Degree Approximations of Polyhedra

Chandrajit L. Bajaj

Jindon Chen

Guoliang Xu

Report Number:

94-002

SMOOTH LOW DEGREE APPROXIMATIONS OF POLYHEDRA

Chandrajit L. Bajaj
Jindon Chen
Guoliang Xu

CSD-TR-94-002
January 1994

Smooth Low Degree Approximations of Polyhedra*

Chandrajit L. Bajaj   Jindon Chen   Guoliang Xu

Department of Computer Sciences
Purdue University
West Lafayette, Indiana

{bajaj,jdc,xuguo}@cs.purdue.edu

Abstract

We present efficient algorithms to construct both C^1 and C^2 smooth meshes of cubic and quintic A-patches to approximate a given polyhedron $\mathcal{P}$ in three dimensions. The A-patch is a smooth and single-sheeted zero-contour patch of a trivariate polynomial in Bernstein-Bezier (BB) form defined within a tetrahedron. The smooth mesh constructions rely on a novel scheme to build an inner simplicial hull Σ consisting of tetrahedra and defined by the faces of the given polyhedron $\mathcal{P}$. A single cubic or quintic A-patch is then constructed within each tetrahedron of the simplicial hull Σ with the resulting surface being C^1 or C^2 smooth, respectively. The free parameters of each individual A-patch can be independently controlled to achieve both local and global shape deformations and a family of C^1 or C^2 smooth approximations of the original polyhedron.

1 Introduction

In this paper, we present efficient algorithms to construct both a C^1 smooth mesh with cubic A-patches and C^2 smooth mesh with cubic and quintic A-patches to approximate a given polyhedron $\mathcal{P}$ in three dimensions. The A-patch is a smooth and single-sheeted zero-contour patch of a trivariate polynomial in Bernstein-Bezier (BB) form defined within a tetrahedron[BCX93], where "A" stands for algebraic. Solutions to the problem of constructing a C^1 mesh of implicit algebraic patches which *interpolate* the vertices of a *simplicial* polyhedron $\mathcal{P}$ have been given by [Dah89] using quadric patches, [BCX93, DTS93, Guo91b, Guo93] using cubic patches and [BI92b] using quintic for convex $\mathcal{P}$(all faces are triangular) and degree seven patches for arbitrary $\mathcal{P}$. While papers [BI92b, Dah89, DTS93, Guo91b, Guo93] provide heuristics based on monotonicity and least square approximation to circumvent the multiple sheeted and singularity problems of implicit patches, [BCX93] introduces new sufficiency conditions for the BB form of trivariate polynomials within a tetrahedron, such that the zero contour of the polynomial is a single sheeted non-singular surface within the tetrahedron (the A-patch) and guarantees that its cubic-mesh complex for $\mathcal{P}$ is both

*This work was supported in part by NSF grants CCR 92-22467, DMS 91-01424, AFOSR grants F49620-93-10138, F49620-94-1-0080, NASA grant NAG-1-1473 and a gift from AT&T.

nonsingular and single sheeted. In this paper we use these cubic A-patches to provide a C^1 smooth single sheeted mesh which approximates a given polyhedron. We also present a new scheme for building a C^2 patch complex with quintic surface patches and furthermore do not require the polyhedron $\mathcal{P}$ to be simplicial.

The C^1 interpolation schemes of [BCX93, Dah89, DTS93, Guo91b, Guo93] all build an outside simplicial hull (consisting of a series of edge and face tetrahedra) containing the given polyhedron $\mathcal{P}$. Such a simplicial hull is nontrivial to construct for arbitrary $\mathcal{P}$ (even convex $\mathcal{P}$ with sharp corners) and can give rise to several exceptional situations and degeneracies (co-planarity, hull self-intersection, etc).The new corner-cutting, inner simplicial hull construction of this paper and can handle all convex $\mathcal{P}$ and also arbitrary polyhedra with non-convex faces. This new simplicial hull scheme is the three dimensional generalization of the two-dimensional corner-cutting scheme used to construct C^k continuous bivariate A-splines [BX92].

Related papers which approximate scattered data using implicit algebraic patches are [Baj92, BBX94, BI92a, BIW93, MW91, Pra87, Sed90] and a classification of data fitting using parametric surface patches is given in [Pet90].

The rest of this paper is as follows. Section 2 gives some preliminary facts about Bernstein-Bezier (BB) representations, A-patches and the geometry of simple polyhedra. Section 3 provides a formal definition of a smoothable simplicial hull. Section 4 presents a simplicial hull construction scheme for a simple polyhedron $\mathcal{P}$ with convex faces, where every vertex normal is "above" the faces surrounding the vertex. Section 5 extends the hull construction algorithm to include a simple polyhedron with nonconvex faces. Section 6 presents details of the C^1 and C^2 continuity schemes for cubic and quintic A-patches. Section 7 exhibits the capabilities of local and global shape control of the A-patch approximating mesh under the current construction scheme. Finally, Section 8 provides some implementation details.

2 Notation and Preliminary Details

2.1 Bernstein-Bezier Representation and A-Patches

Let $\{p_1, \cdots p_j\} \in I\!R^3$. Then the *convex hull* of these points is defined by $[p_1p_2 \ldots p_j] = \{p \in I\!R^3 : p = \sum_{i=1}^{j} \alpha_i p_i, \alpha_i \geq 0, \sum_{i=1}^{j} \alpha_i = 1\}$. Let $p_1,\ p_2,\ p_3,\ p_4 \in I\!R^3$ be affine independent. Then the tetrahedron(or three dimensional simplex) with vertices p_1, p_2, p_3, and p_4, is $V = [p_1p_2p_3p_4]$. For any $p = \sum_{i=1}^{4} \alpha_i p_i \in V$, $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)^T$ is the barycentric coordinate of p. Let $p = (x, y, z)^T$, $p_i = (x_i, y_i, z_i)^T$. Then the barycentric coordinates relate to the Cartesian coordinates via the following relation

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} \tag{2.1}$$

Any polynomial $f(p)$ of degree m can be expressed in Bernstein-Bezier(BB) form over V as $f(p) = \sum_{|\lambda|=m} b_\lambda\ B_\lambda^m(\alpha),\quad \lambda \in Z_+^4$ where $B_\lambda^m(\alpha) = \frac{m!}{\lambda_1!\lambda_2!\lambda_3!\lambda_4!}\ \alpha_1^{\lambda_1}\alpha_2^{\lambda_2}\alpha_3^{\lambda_3}\alpha_4^{\lambda_4}$ are the trivariate Bernstein polynomials for $|\lambda| = \sum_{i=1}^{4} \lambda_i$ with $\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)^T$. Also $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)^T$ is the barycentric coordinate of p, $b_\lambda = b_{\lambda_1\lambda_2\lambda_3\lambda_4}$(as a subscript, we simply write λ as $\lambda_1\lambda_2\lambda_3\lambda_4$) are

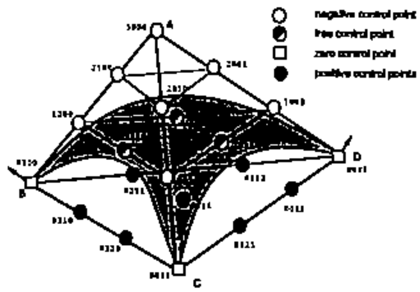


Figure 2.1: A Smooth and Single-Sheeted Triangular Algebraic Surface Patch(A triangular A-patch)

called control points, and $\mathcal{Z}_+^4$ stands for the set of all four dimensional vectors with nonnegative integer components. Let

$$F(\alpha) = \sum_{|\lambda|=m} b_\lambda B_\lambda^m(\alpha), \ |\alpha| = 1, \tag{2.2}$$

be a given polynomial of degree m on the tetrahedron $S = \{(\alpha_1, \alpha_2, \alpha_3, \alpha_4)^T \in \mathbb{R}^4 \ : \ \sum_{i=1}^4 \alpha_i = 1, \ \ \alpha_i \geq 0\}$. The surface patch within the tetrahedron is defined by $S_f \subset S \ : \ F(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = 0$.

Definition 2.1 *Triangular algebraic surface patch*

If any line segment passing through the j-th vertex v_j of S and its opposite face $S_j = \{(\alpha_1, \alpha_2, \alpha_3, \alpha_4)^T : \alpha_j = 0, \ \ \alpha_i > 0, \ \sum_{i \neq j} \alpha_i = 1\}$ intersects S_f only once, then we call S_f a triangular j-patch (see Figure 2.1).

Note, for a given tetrahedron, we can have triangular j-patches for $j = 1, 2, 3, 4$. All these patches are A-patches (algebraic patches).

Lemma 2.1 *The triangular j-patch is smooth (non-singular) and single-sheeted.*

Proof. See [BCX93]. $\diamond$

Theorem 2.1 *Let $F(\alpha)$ be defined as (2.2), and $j (1 \leq j \leq 4)$ be a given integer. If there exists an integer $k (0 < k < m)$ such that*

$$b_{\lambda_1\lambda_2\lambda_3\lambda_4} \geq 0, \qquad \lambda_j = 0, 1, \ldots, k-1, \tag{2.3}$$

$$b_{\lambda_1\lambda_2\lambda_3\lambda_4} \leq 0, \qquad \lambda_j = k+1, \ldots, m. \tag{2.4}$$

and $\sum_{\substack{|\lambda|=m \\ \lambda_j=0}} b_\lambda > 0$, $\sum_{|\lambda|=m} b_\lambda < 0$ for at least one $\lambda_j (k < \lambda_j \leq m)$. Then S_f is a triangular j-patch.

Proof. See [BCX93]. $\diamond$

Lemma 2.2 *Let $f(p) = \sum_{|\lambda|=n} b_\lambda B_\lambda^n(\alpha)$ be defined on the tetrahedron $[p_1 p_2 p_3 p_4]$, then*

$$b_{(n-1)e_i+e_j} = b_{ne_i} + \frac{1}{n}(p_j - p_i)^T \nabla f(p_i), \quad j = 1, 2, 3, 4; \quad j \neq i \tag{2.5}$$

$$
\begin{aligned}
b_{(n-2)e_i+e_j+e_k} &= -b_{ne_i} + b_{(n-1)e_i+e_j} + b_{(n-1)e_i+e_k} \\
&+ \tfrac{1}{n(n-1)}(p_j - p_i)^T \nabla^2 f(p_i)(p_k - p_i), \quad j \neq i, k \neq i
\end{aligned} \tag{2.6}
$$

(2.5) can be found in [Guo91a](p.23). (2.6) is derived from directional derivative formulas (see [Far90] p.310).

Lemma 2.3 *([Far90] p.318) Let $f(p) = \sum_{|\lambda|=n} a_\lambda B_\lambda^n(\alpha)$ and $g(p) = \sum_{|\lambda|=n} b_\lambda B_\lambda^n(\alpha)$ be two polynomials defined on two tetrahedra $[p_1p_2p_3p_4]$ and $[p_1'p_2p_3p_4]$, respectively. Then*
(i) f and g are C^0 continuous at the common face $[p_2p_3p_4]$ if and only if

$$a_\lambda = b_\lambda, \quad for \quad any \quad \lambda = 0\lambda_2\lambda_3\lambda_4, \quad |\lambda| = n \tag{2.7}$$

(ii) f and g are C^1 continuous at the common face $[p_2p_3p_4]$ if and only if (2.7) holds and

$$b_{1\lambda_2\lambda_3\lambda_4} = \beta_1 a_{1\lambda_2\lambda_3\lambda_4} + \beta_2 a_{0\lambda_2\lambda_3\lambda_4+0100} + \beta_3 a_{0\lambda_2\lambda_3\lambda_4+0010} + \beta_4 a_{0\lambda_2\lambda_3\lambda_4+0001} \tag{2.8}$$

(iii) f and g are C^2 continuous at the common face $[p_2p_3p_4]$ if and only if (2.7)-(2.8) holds and

$$
\begin{aligned}
b_{1\lambda_2\lambda_3\lambda_4} &= \beta_1^2 a_{2\lambda_2\lambda_3\lambda_4} + 2\beta_1\beta_2 a_{0\lambda_2\lambda_3\lambda_4+1100} + 2\beta_1\beta_3 a_{0\lambda_2\lambda_3\lambda_4+1010} + 2\beta_1\beta_4 a_{0\lambda_2\lambda_3\lambda_4+1001} \\
&+ \beta_2^2 a_{0\lambda_2\lambda_3\lambda_4+0200} + 2\beta_2\beta_3 a_{0\lambda_2\lambda_3\lambda_4+0110} + 2\beta_2\beta_4 a_{0\lambda_2\lambda_3\lambda_4+0101} \\
&+ \beta_3^2 a_{0\lambda_2\lambda_3\lambda_4+0020} + 2\beta_3\beta_4 a_{0\lambda_2\lambda_3\lambda_4+0011} + \beta_4^2 a_{0\lambda_2\lambda_3\lambda_4+0002}
\end{aligned} \tag{2.9}
$$

where $\beta = (\beta_1, \beta_2, \beta_3, \beta_4)^T$ are defined by the following relation

$$p_1' = \beta_1 p_1 + \beta_2 p_2 + \beta_3 p_3 + \beta_4 p_4, \quad |\beta| = 1$$

In Lemma 2.3, if $\beta_2 = \beta_3 = 0$, that is p_1', p_4 and p_1 are collinear, then (2.8) and (2.9) become

$$a_{0\lambda_2\lambda_3\lambda_4+0001} = \mu_1 a_{1\lambda_2\lambda_3\lambda_4} + \mu_2 b_{1\lambda_2\lambda_3\lambda_4} \tag{2.10}$$

$$\mu_1^2 a_{2\lambda_2\lambda_3\lambda_4} - \mu_1 a_{0\lambda_2\lambda_3\lambda_4+1001} = \mu_2^2 b_{2\lambda_2\lambda_3\lambda_4} - \mu_2 b_{0\lambda_2\lambda_3\lambda_4+1001} \tag{2.11}$$

respectively, where $\mu_1 = -\frac{\beta_1}{\beta_4}, \mu_2 = \frac{1}{\beta_4}$, that is $p_4 = \mu_1 p_1 + \mu_2 p_1'$.

2.2 Geometry of Simple Polyhedra

Definition 2.2 *A point p is* separable *from other points of interest if there exists a plane π that separates p and the other points. It is* unseparable *from them otherwise.*

Definition 2.3 *The* edge angle *of edge E of a polyhedron is inner dihedral angle between the two faces incident at E*

Definition 2.4 *On a polyhedron $\mathcal{P}$, an edge E is a* ridge *if its edge angle is less than or equal to π; E is a* valley *otherwise.*

Definition 2.5 *For a vertex V of a planar polygon, a* corner guard G_V *of V is a point in the interior Q from where corner V is visible, namely, the two edges incident to V are visible. A* corner guard set *is a set of corner guards from which all the corners are visible. A* corner guard net *is a connected planar graph inside Q such that any corner V is visible from at least one of its vertices.*

For a convex polygon or a star-shaped polygon, a minimum corner guard set consists of just a single point. Especially for a convex polygon, any point in the interior polygon forms a minimum (singleton) corner guard set. Figure 2.2 illustrates different kinds of polygons and their corner guard regions.

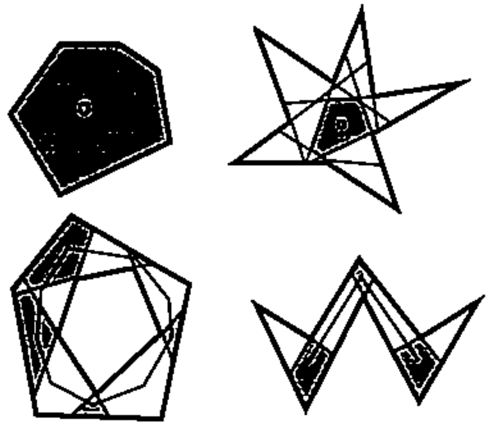


Figure 2.2: Polygons and Their Corner Guard Net. Shaded parts are possible regions to position guards

3 Smoothable Simplicial Hull

Definition 3.1 *Let $[p_ip_j]$ be an edge of a polyhedron $\mathcal{P}$ with endpoint vertex normals n_i and n_j. If $(p_j - p_i)^T n_i \; (p_i - p_j)^T n_j \geq 0$, then the edge is* convex. *Otherwise, it is* nonconvex. *If the edge satisfies the convex condition. and at least one of $(p_j - p_i)^T n_i$ and $(p_i - p_j)^T n_j$ is positive, then we say the edge $[p_ip_j]$ is* positively convex. *If both of them are zero then we say it is* zero convex. *If at least one of them is negative, the edge is* negatively convex.

Definition 3.2 *Let $[p_ip_jp_k]$ be a triangular face of a polyhedron $\mathcal{P}$. If its three edges are non-negatively (positively or zero) convex and at least one of them is positive convex, then we say the face $[p_ip_jp_k]$ is* positively convex. *If all the three edges are zero convex then the face is* zero convex. *If its three edges are nonpositively (negatively or zero) convex and at least one of them is negatively convex, the face is* negatively convex. *Otherwise, $[p_ip_jp_k]$ is* non-convex.

Note, that here we are overloading the term *convex* to characterize the relations between the vertex normals and edges of faces. We distinguish between convex and non-convex faces in the simplicial hull below where we build one tetrahedra for convex faces and double tetrahedra for non-convex faces.

Definition 3.3 *A polyhedron $\mathcal{P}$ with vertex normals is* edge-convex *if every edge is convex.*

Definition 3.4 *A* face-tetrahedron $[p_ip_jp_kq_l]$ *is a tetrahedron that is built based on a triangular face $[p_ip_jp_k] \in \mathcal{P}$. A face-tetrahedron $[p_ip_jp_kq_l]$ is* {positively | zero | negative | non-)} convex *if the face $[p_ip_jp_k]$ is {positively | zero | negative | non-)} convex. A face-tetrahedron is* U-nonconvex*(U for upper) if (1) it is outside $\mathcal{P}$ and has one positively convex edge; or (2) it is inside $\mathcal{P}$ and has one negatively convex edge. A nonconvex face-tetrahedron is* L-nonconvex*(L for lower) if (1)it is outside the triangulation $\mathcal{P}$ and has two positively convex edge; or (2) it is inside $\mathcal{P}$ and has two negatively convex edge.*

Definition 3.5 *A convex face-tetrahedron* $[p_1p_2p_3p_4]$ *is* tangent-containing *if the tangent planes at the three interpolatory vertices* p_1, p_2 *and* p_3 *are tangent with* $[p_1p_2p_3p_4]$; *A pair of LU face-tetrahedra* $[p_1p_2p_3p_4q_4]$ *is* tangent-containing *if the tangent planes at the three interpolatory vertices* p_1, p_2 *and* p_3 *are tangent with either* $[p_1p_2p_3p_4]$ *or* $[p_1p_2p_3q_4]$;

The term *simplicial hull* is loosely defined as a contiguous collection of tetrahedra(simplices) constructed on a base simplicial polyhedron. We define one kind of simplicial hull that we shall use here and refer to it as *smoothable simplicial hull.*

Definition 3.6 *A* smoothable simplicial hull $\Sigma = (\mathcal{T}, \mathcal{S}_f, \mathcal{S}_e, \mathcal{R}_{tf}, \mathcal{R}_{ffe})$ *where*
(1) $\mathcal{T} = [p_ip_jp_k]$ *is an edge-convex triangulation;*
(2) $\mathcal{S}_f = [p_ip_jp_kq_l]$ *is a collection of face tetrahedra, where* $[p_ip_jp_k] \in \mathcal{T}$;
(3) $\mathcal{S}_e = [p_ip_jq_ks_l]$ *is a collection of edge tetrahedra pairs, where* $[p_ip_j] \in \mathcal{T}$ *and* $[p_ip_jq_k] \in \mathcal{S}_f$;
(4) $\mathcal{R}_{tf} = \mathcal{T} \times \mathcal{S}_f$ *is a relation between* $\mathcal{T}$ *and* $\mathcal{S}_f$, *which can be described as (i) (single sided) there is one* tangent-plane-containing *face-tetrahedron* $[p_ip_jp_kq_l] \in \mathcal{S}_f$ *is built on a convex face* $[p_ip_jp_k] \in \mathcal{T}$ *and (ii) (double sided) there are a* tangent-plane-containing *pair of LU-face-tetrahedra* $[p_ip_jp_kq_l]$, $[p_ip_jp_k\overline{q_l}] \in \mathcal{S}_f$ *are built on a nonconvex face* $[p_1p_2p_3] \in \mathcal{T}$, *one on each side;*
(5) $\mathcal{R}_{ffe} = \mathcal{S}_f \times \mathcal{S}_f \times \mathcal{S}_e$ *is a relation between face-tetrahedra and a pair of edge tetrahedra, which can be described as (i) (non-intersection) two face-tetrahedra* $[p_ip_jp_kq_l]$, $[p_ip_jp_mq_n] \in \mathcal{S}_f$ *that share a common edge* $[p_ip_j] \in \mathcal{T}$ *does not intersect each other and (ii) a pair of edge-tetrahedra* $[p_ip_jq_ls_r]$, $[p_ip_jq_ns_r] \in \mathcal{S}_e$ *where* $s_r = \frac{(q_l+q_n)}{2}$, *are built between the pair of edge-sharing face-tetrahedra that are both inside or both outside the triangulation* $\mathcal{T}$.

4 Simple Polyhedron with Convex Faces and "Above Face" Vertex Normals

In this section, we present an algorithm for constructing a smoothable simplicial hull from a simple polyhedron $\mathcal{P}$ with three restrictions. (i) every face is convex. (ii) every vertex normal is "above" the incident faces, namely for every vertex V. A vertex normal n_V is above its incident faces f_i if the inner product of n_V with all the face normals of f_i is positive. The vertex normal can be computed in different ways such as face normals or local interpolation by a sphere [BI92b, BCX93, Pet90]. (iii) the polyhedron is manifold (i.e. two faces incident per edge,etc)

Algorithm 1 *INPUT: A simple polyhedron* $\mathcal{P}$ *with convex faces and "above face" vertex normals.*

(1) Compute the centroid C_i *of each face* f_i.

(2) For every vertex V, *if it is a trihedral vertex, go to (3); otherwise go to (4). Go to (6) if every vertex is done.*

(3) Build face-tetrahedron $[C_1C_2C_3V]$, *where* C_i *is the centroids of a face* $f_i \in \mathcal{P}$ *around* V. *(See Figure 4.3) Go back to (3) for the next vertex.*

(4) Let $l_V = V + n_Vt$, *be a straight line pass through vertex* V *in the direction of the normal* n_V. *Compute the projection* p_{c_i} *on* l_V *of each of the centroids* C_i *of the faces* f_i *around* V. $(p_{C_i} = V + n_Vt_i)$. *If* $t_i > 0$ *for all* i *or* $t_i < 0$ *for all* i, *then* V *is separable from the centroids* C_i *of the faces* f_i *around* V, *by a plane* π *parallel to the tangent plane at* V, *otherwise* V *is unseparable. In the case of a separable vertex* V, *let* $\overline{V}$ *be the projection closest to* V, *or the one with the smallest* t_i *in absolute value;* $(\overline{V} = V + n_Vt^*$, *where* $t^* \in \{t_i\}$ *and* $|t^*| = min\{|t_i|\})$. *In the case of an*

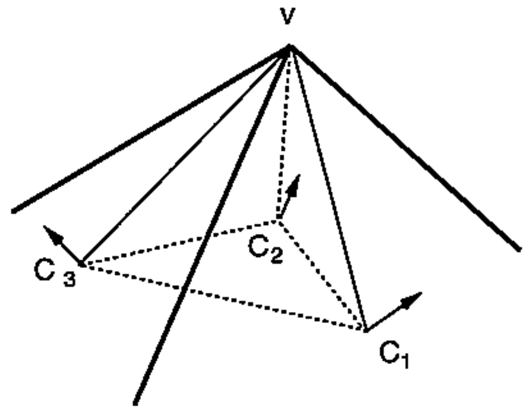


Figure 4.3: Construct a Face-tetrahedron on a Trihedral Vertex

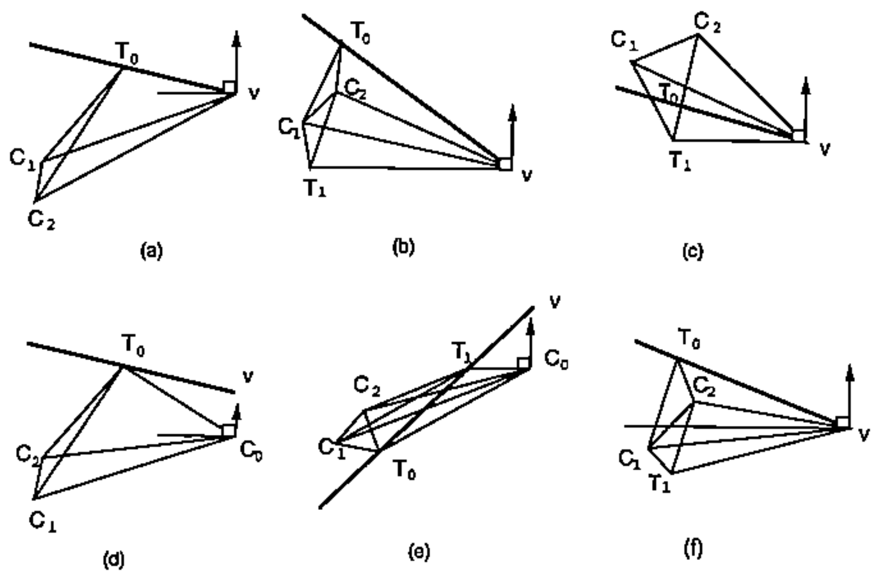


Figure 4.4: Construction of Face-tetrahedron on a Non-trihedral Vertex

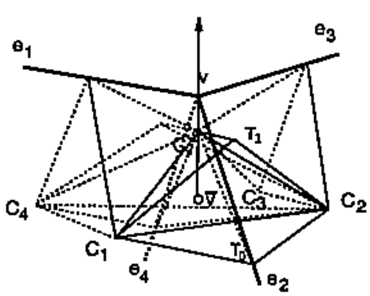


Figure 4.5: Face-tetrahedron around a Non-trihedral Vertex

unseparable vertex V, let $\overline{V} = V$. Let C_0, denote a pseudo centroid, *and be a point between V and $\overline{V}$. ($C_0 = \alpha V + (1-\alpha)\overline{V}$ for some $\alpha \in (0,1)$). Let n_V be the normal at C_0. Go to (5).*

(5) Let $e = VV'$ be an edge incident to V and C_1, C_2 be the two centroids of the two faces incident at edge e. Let T_0 is the middle point between the intersection point of π_{C_0} and M, the middle point of edge e. Let T_1 be the projection of $M_{VV'}$, the midpont of VV' on π_{C_0}, the tangent plane of C_0. Now we construct the face-tetrahedra based on the signs of the inner products between the normal n_{C_0} and C_1, C_2, T_0 and T_1. (i) $(n_{C_0}C_1)(n_{C_0}C_2) \geq 0$, namely the two inner product are of one sign, and $(n_{C_0}T_0)(n_{C_0}C_1) \leq 0$. Build a face-tetrahedron $[C_0C_1C_2T_0]$. (See Figure 4.4 (a) (d).) (ii) $(n_{C_0}C_1)(n_{C_0}C_2) \geq 0$, $(n_{C_0}T_0)(n_{C_0}C_1) > 0$, T_0 and T_1 are on one side of $[C_0C_1C_2]$. Build face-tetrahedron $[T_1C_0C_1C_2]$. (See Figure 4.4(c).) (iii) $(n_{C_0}C_1)(n_{C_0}C_2) \geq 0$, $(n_{C_0}T_0)(n_{C_0}C_1) > 0$, T_0 and T_1 are on different sides of $[C_0C_1C_2]$. Build L-U-face-tetrahedra $[T_0C_0C_1C_2]$ and $[T_1C_0C_1C_2]$. One on each side of $[C_0C_1C_2]$. (See Figure 4.4(b) and (e).) (iv) $(n_{C_0}C_1)(n_{C_0}C_2) < 0$. Assume that T_0 is "above" $[T_1C_0C_1C_2]$, namely e is a ridge. Move T_0 along the direction of n_{C_0} until it is "above" π_{C_0}, namely $(n_{C_0}T_0) > 0$. Move T_1 along the opposite direction of n_{C_0} until it is "above" π_{C_0}, namely $(n_{C_0}T_1) < 0$. If e is a valley, move T_0 and T_1 in opposite directions. Build L-U-face-tetrahedra $[T_0C_0C_1C_2]$ and $[T_1C_0C_1C_2]$. (See Figure 4.4(f).) (Figure 4.5 illustrates that 4 face-tetrahedra are built around a vertex with 4 edges.) Go back to (3) for the next vertex.

(6) Build one pair of edge-tetrahedra between each pair of adjacent face-tetrahedra of the same orientation. (See Figure 6.9. Forget about the nodes on the tetrahedra for the moment.)

OUTPUT: A simplicial hull Σ.

Let $\mathcal{T}$ be the triangulation consists of the triangular faces $[C_1C_2C_3]$'s and $[C_0C_1C_2]$'s, built in step (3) and step (5) of Algorithm 1, with respect to different vertices and edges. Let $\mathcal{S}_f$ be the collection of face-tetrahedra and $\mathcal{S}_e$ be the collection of edge-tetrahedra. Conditions (2), (3) and (5)(ii) of definition 3.6 are obviously satisfied by the construction of Σ. Along with the following lemmas, we conclude that Σ is a smoothable simplicial hull. See Appendix A for details and proofs of the following lemma.

Lemma 4.1 *Triangulation $\mathcal{T}$ is edge-convex.*

Lemma 4.2 *Every face-tetrahedron $[p_ip_jp_kq_l] \in \mathcal{S}_f$ is tangent-containing.*

Lemma 4.3 *The face-tetrahedra in $\mathcal{S}_f$ do not intersect each other.*

Theorem 4.1 *Σ is a smoothable simplicial hull.*

For a simple polyhedron that does not satisfy the three restrictions, certain preprocessing steps can be taken to enforce them. Appendix C gives an algorithm to transform an arbitrary polyhedron to one that has "above face" vertex normals. The convex face restriction can be enforced by subdividing nonconvex faces into convex ones. However, the next section discusses a modified version of Algorithm 1 by which a simplicial hull can be constructed directly on a simple polyhedron with "above face" vertex normals, which could have nonconvex faces.

In [BCX93], extra subdivision are needed if the adjacent triangular faces of the triangulation $\mathcal{T}$ are coplanar to each other. Otherwise, all the bottom weights of the coplanar face-tetrahedra are related to each other by the continuity constraints so that the locality property of the weight-setting

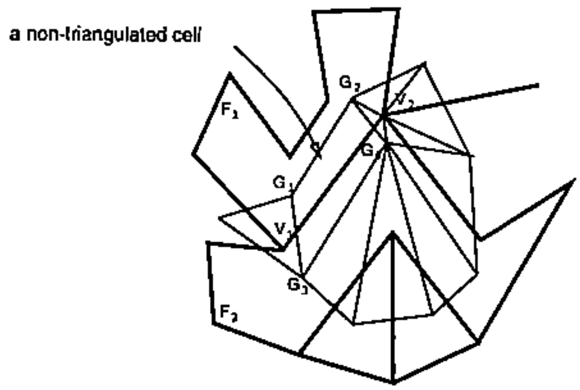


Figure 5.6: Triangulating a Mesh of Corner Guard Nets and Nontrihedral vertices

procedure is destroyed and also the single sheeted condition is jeopardized. For the smoothable simplicial hull scheme presented in this section, as the triangulation of the polyhedron is also constructed instead of being given. By selecting "good" centroid combinations, one can construct a hull without any coplanar faces or only with some "trivial" coplanar cases. See Appendix B for details.

5 Extending the Algorithm to Include Polyhedra with Nonconvex Faces

In this section, we extend our Algorithm 1 to allow nonconvex facets in polyhedron $\mathcal{P}$.

Construction of the face-tetrahedron requires that the corner is visible from the centroid. While every point in a convex polygon meets this requirement, it is not the case for a nonconvex polygon. For each nonconvex polygon, we first construct a corner guard net (Definition 2.5), and set the vertex normal of the corner guards as the face normal. For each convex polygon, take a centroid(as we did in last section) to be the corner guard. For a vertex V, connect the neighboring corner guards with respect to vertex V. If V is non-trihedral, connect C_V, pseudo centroid of V and the surrounding corner guards. Now the corner guards and the pseudo centroids C_V's,are connected by the corner guard net and the newly added edges, to form a mesh Q^* with non-triangular cells (due to multiple corner guards in some nonconvex faces. see Figure 5.6). Triangulating the four-sided non-triangular cells, we obtain a triangulation Q. By similar arguments as in the last section, the edges between corner guards of different faces, and the edges between corner guards and pseudo centroids are all convex (Definition 3.1) And the edges in corner guard nets are zero convex by construction. Hence Q is edge-convex.

There are some nonconvex corners with respect to a nonconvex polygonal face. One example is shown in Figure 5.7. Here two face-tetrahedra are needed although the vertex is trihedral.

Now we extend Algorithm 1 by replacing the centroids of non-convex faces by the corner guards.

Algorithm 2 *INPUT:A simple polyhedron $\mathcal{P}$ with "above face" vertex normals.*

(1) compute the corner guard net of each faces.

(2) For every vertex V, if it is a trihedral vertex and each pair of edges form an angle $\leq \pi$, go to (3); if V is a trihedral vertex and one pair of edges e_2 and e_3 form an angle greater than π, go

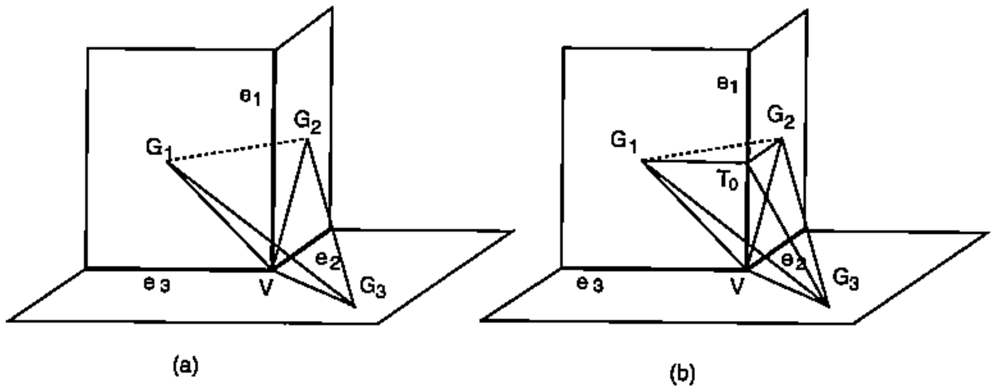


Figure 5.7: A Hallway Corner

to (3'); Otherwise go to (4). Go to (5') if every vertex is done.

(3') Build two face-tetrahedra $[G_1G_2G_3V]$ and $[G_1G_2G_3T_0]$, where T_0 is a point between M_{e_1}, the middle point of e_1, and the intersection of e_1 and $[G_1G_2G_3V]$. (see Figure 5.7)

(3) (4) (5) (replace centroids by corner guards).

(5') (See Figure 5.6) For each pair of adjacent face f_1 and f_2, let edge V_1V_2 be the edge they share, G_1 and G_2 be the corner guards of f_1 with respect to vertices V_1 and V_2, G_3 and G_4 be the corner guards of f_2 with respect to vertices V_1 and V_2. If $G_1 \neq G_2$ and $G_3 \neq G_4$, build face-tetrahedra $[G_1G_2G_3H_0]$ and $[G_2G_3G_4H_1]$, where H_0 is a vertex between $M_{V_1V_2}$, the midpoint of V_1V_2, and top vertex of the face-tetrahedron shares edge $[G_1G_3]$ that was built in Step (3) or (5) of the algorithm, similarly, H_1 is on the other side of $M_{V_1V_2}$. If $G_1 = G_2$ and $G_3 \neq G_4$, build face-tetrahedron $[G_1G_3G_4M_{V_1V_2}]$. If $G_1 \neq G_2$ and $G_3 = G_4$, build face-tetrahedron $[G_1G_2G_3M_{V_1V_2}]$.

(6) (same as in Algorithm 1)

OUTPUT: A simplicial hull Σ.

It is not difficult to show that simplicial hull Σ is also smoothable.

6 C^1 Mesh of Cubic Patches and C^2 Mesh of Quintic Patches

Once we have established a smoothable simplicial hull $\sum$ for the given polyhedron $\mathcal{P}$ and a set of point normals N, we construct a C^1 or C^2 trivariate piecewise polynomial function f within $\sum$ such that f has the given C^1 or C^2 data at each vertex and the zero contour of f within $\sum$ form a C^1 or C^2 continuous surface with the same topology as T. We adopt the C^1 cubic scheme from [BCX93] using only the special cases needed for the new smoothable simplicial hull $\sum$.

For the construction of f within the tetrahedra built on two adjacent triangles(see Figure 6.8 for C^1 and Figure 6.9 for C^2. See also Figure 7.10 for examples of the C^1 and C^2 surfaces), let

$$V_1 = [p_1p_2p_3p_4], \quad V_2 = [p_1'p_2p_3p_4'], \quad W_1 = [p_1''p_2p_3p_4]$$
$$W_2 = [p_1''p_2p_3p_4'], \quad V_1' = [p_1p_2p_3q_4], \quad V_2' = [p_1'p_2p_3q_4']$$

and the polynomials f_i over V_i, g_i over W_i and f_i' over V_i' be expressed in Bernstein-Bezier forms with coefficients a_λ^i, b_λ^i and c_λ^i, respectively. Now we shall determine these coefficients step by step.

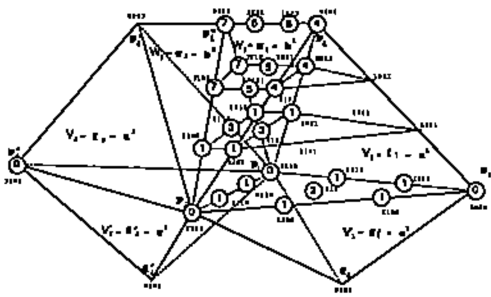


Figure 6.8: Adjacent Tetrahedra, Cubic Functions and Control Points for two Non-Convex Adjacent Faces

Denote

$$\begin{array}{rcll} p_1'' & = & \beta_1^1 p_1 + \beta_2^1 p_2 + \beta_3^1 p_3 + \beta_4^1 p_4, & \beta_1^1 + \beta_2^1 + \beta_3^1 + \beta_4^1 = 1 \\ p_1'' & = & \beta_1^2 p_1' + \beta_2^2 p_2 + \beta_3^2 p_3 + \beta_4^2 p_4', & \beta_1^2 + \beta_2^2 + \beta_3^2 + \beta_4^2 = 1 \\ p_1'' & = & \mu_1 p_4 + \mu_2 p_4', & \mu_1 + \mu_2 = 1 \end{array} \tag{6.12}$$

C^1 cubic scheme(see Figure 6.8)

1. The number 0 weights are given by the function values at the vertices.
2. The number 1 weights are determined by formula (2.5).
3. The number 2 weights, that is $a_{1110}^{(i)}$, are free.
4. The number 3 weights are determined by C^1 conditions (2.8).
5. The number 4 weights are free.
6. The number 5 weights are determined by C^1 conditions (2.8).
7. The number 6 weights are free.
8. The number 7 weights are determined by C^1 conditions (2.10).

   The remaining weights with index $\lambda_1\lambda_2\lambda_3\lambda_4$ are determined by C^1 condition (2.8) for $\lambda_4 \le 1$ and freely chosen for $\lambda_4 > 1$.

C^2 quintic scheme(see Figure 6.9)

1. The number 0 weights are given by the function values at the vertices. For examples, $a_{5e_i}^{(1)} = f(p_i)$, $i = 1, 2, 3$.
2. The number 1 weights are determined by formula (2.5).
3. The number 2 weights are determined by formula (2.6).
4. The number 3 weights, that is $a_{1220}^{(i)}, a_{2210}^{(i)}$ and $a_{2120}^{(i)}$, are free.

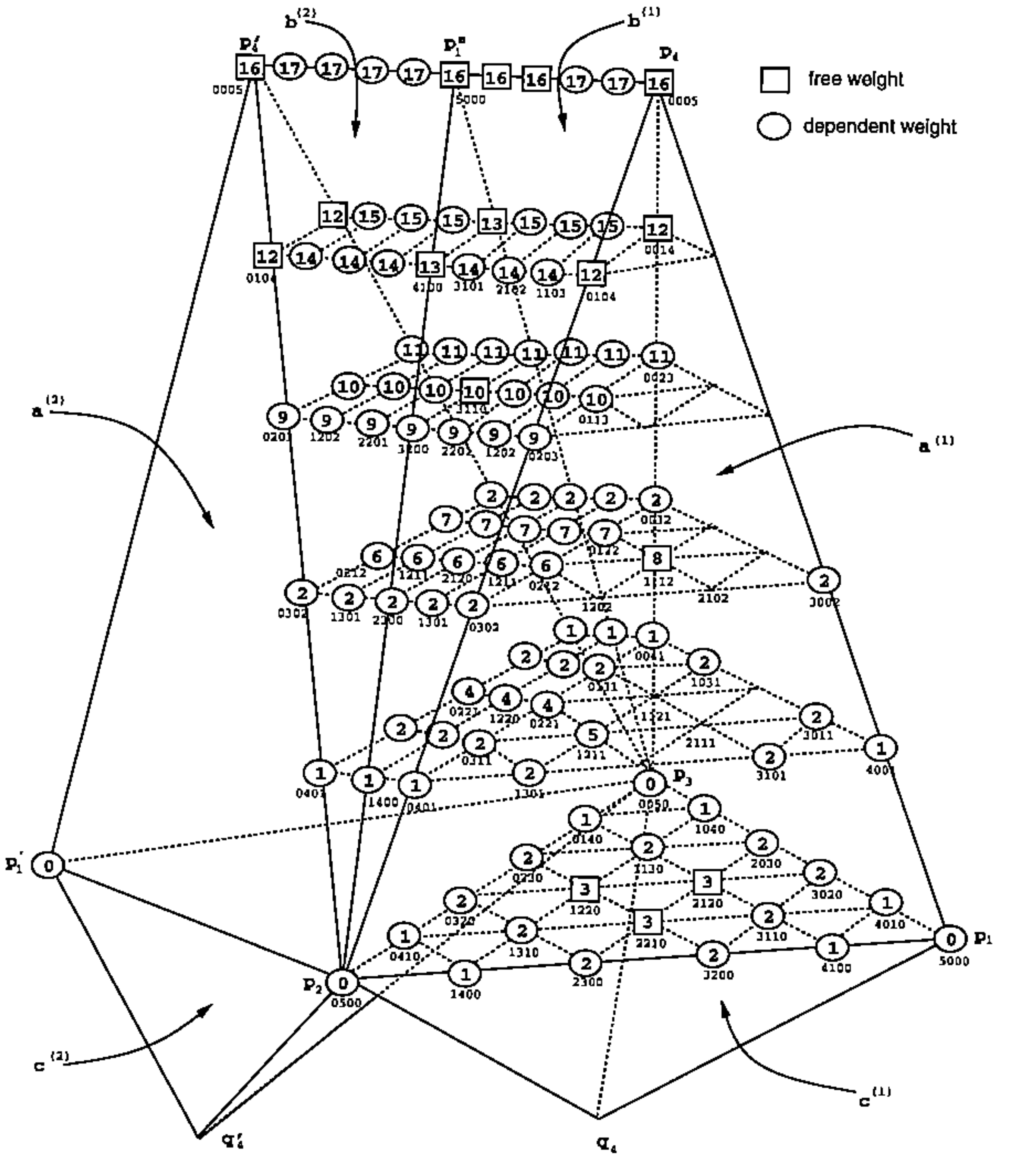


Figure 6.9: Adjacent Tetrahedra, Quintic Functions and Control Points for two Non-Convex Adjacent Faces

5. The number 4 weights are determined by C^1 conditions (2.8), that is

$$b^{(i)}_{1220} = \beta^{(i)}_1 a^{(i)}_{1220} + \beta^{(i)}_2 a^{(i)}_{0320} + \beta^{(i)}_3 a^{(i)}_{0230} + \beta^{(i)}_4 a^{(i)}_{0221},\ i = 1, 2$$

$$b^{(i)}_{1220} = \mu_1 a^{(1)}_{0221} + \mu_2 a^{(2)}_{0221}$$

It follows from these equations that

$$\mu_1 a^{(1)}_{0221} + \mu_2 a^{(2)}_{0221} - \beta^{(i)}_4 a^{(i)}_{0221} = \beta^{(i)}_1 a^{(i)}_{1220} + \beta^{(i)}_2 a^{(i)}_{0320} + \beta^{(i)}_3 a^{(i)}_{0230} \tag{6.13}$$

for $i = 1, 2$. The coefficient matrix A of (6.13) for the unknowns $a^{(i)}_{0221}$ is

$$A = \begin{bmatrix} \mu_1 - \beta^{(1)}_4 & \mu_2 \\ \mu_1 & \mu_2 - \beta^{(2)}_4 \end{bmatrix}$$

This matrix is nonsingular if and only if $p_1, p_1' p_2$ and p_3 are not coplanar [BCX93]. Hence (6.13) has unique solution under our assumptions.

6. The number 5 and 6 weights have to be determined simultaneously. In determining these weights, we need to consider all the C^1 and C^2 conditions related to the tetrahedra surrounding the vertex p_2. Suppose there are k triangles(hence k edges) around p_2 and the convexity change of the edges occurs r times, then by C^1 and C^2 conditions, we have $6k + r$ equations That is, crossing each face, we have two equations, and crossing each triangle, we have one equation. The number of related unknowns is also $6k + r$. That is, k number 5 weights and $5k$ number 6 weights and one more unknown is related when a single convexity change of the edges occur. See Appendix D for further details.

7. The number 7 weights are similarly determined as that of number 6 weights.

8. The number 8 weight $a^{(i)}_{1112}$ are free.

9. The number 9 weights are determined by C^1 and C^2 conditions. Both the number of equations and the number of unknowns are $6k$. See Appendix E for details.

10. For the number 10 weights, we have six equations parallel to the equations (E.23)–(E.26) with all the index changed by the rule

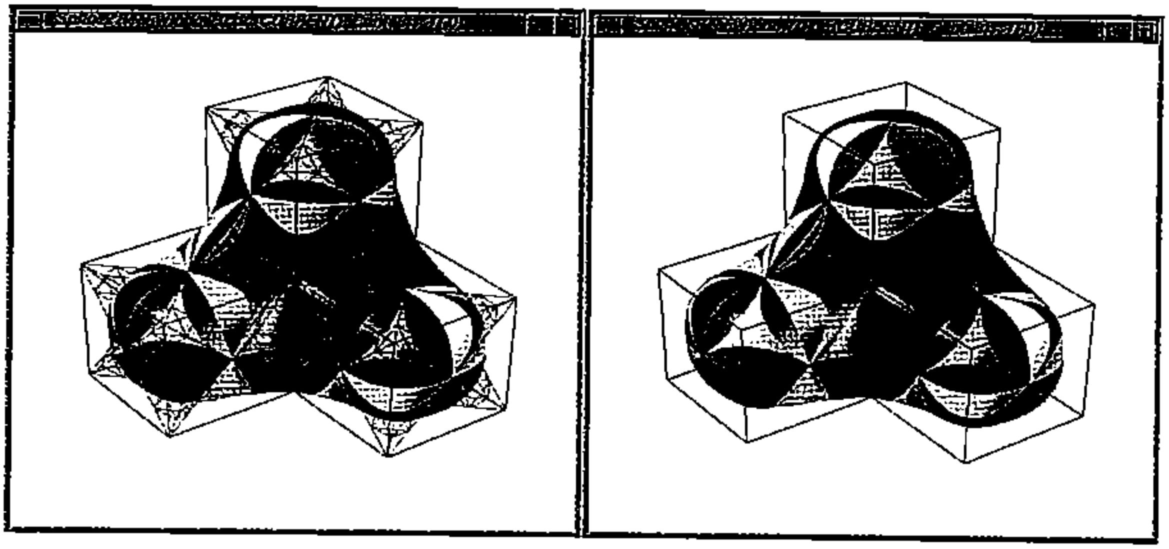
$$\text{the index of the number 10 weright} = \text{the index of the number } 9 - e_2 + e_3 \tag{6.14}$$

and seven independent weights. By chosing one of them, say $b^{(i)}_{3110}$, to be a free parameter, the entire system can be solved.

11. The number 11 weights are determined in the same way as the that of number 9 weights.

12. The number 12 and 13 weights are free. The number 14 weights are determined by C^1 and C^2 conditions. That is $b^{(i)}_{1103}$ are defined by (2.8), and $b^{(i)}_{2102}$ are defined by (2.9). For $b^{(i)}_{3101}$, we have by (2.10) and (2.11)

$$\mu_1 b^{(1)}_{3101} + \mu_2 b^{(2)}_{3101} = b^{(1)}_{4100}$$



(a) (b)

Figure 7.10: C^1 and C^2 Smooth Approximations of a Polyhedron

$$-\mu_1 b^{(1)}_{3101} + \mu_2 b^{(2)}_{3101} = \mu_2^2 b^{(2)}_{2102} - \mu_1^2 b^{(2)}_{2102}$$

Hence

$$b^{(1)}_{3101} = \frac{b^{(1)}_{4100} - \mu_2^2 b^{(2)}_{2102} + \mu_1^2 b^{(2)}_{2102}}{2\mu_1}$$

$$b^{(2)}_{3101} = \frac{b^{(1)}_{4100} + \mu_2^2 b^{(2)}_{2102} - \mu_1^2 b^{(2)}_{2102}}{2\mu_1}$$

13. The number 15 weights are similar to that of number 14, the index is changed by the rule (6.14).

14. The number 16 weights are free, the number 17's are determined by C^1 and C^2 conditions.

15. The remaining weights with index $\lambda_1\lambda_2\lambda_3\lambda_4$ are determined by C^1 and C^2 conditions (2.8) and (2.9) for $\lambda_4 \leq 2$ and freely chosen for $\lambda_4 > 2$

In summary, the construction step 1–14 is according to the C^1 and C^2 conditions across the common tetrahedra faces that are over or below the original triangulation. Step 15 is according to the C^1 and C^2 conditions across the common tetrahedra faces that are on the original triangulation. Therefore, the composite polynomial function is global C^2 continuous.

7 Shape Control

7.1 Cutting the corner by different ratio

We can control how sharply the corners of the original polyhedral are cut. For a trihedral corner that is to be smoothed in step (3) of the algorithm, we cut the corner by different extends to raise or lower the A-patch defined inside the face-tetrahedron. The top of the surface can be as high as

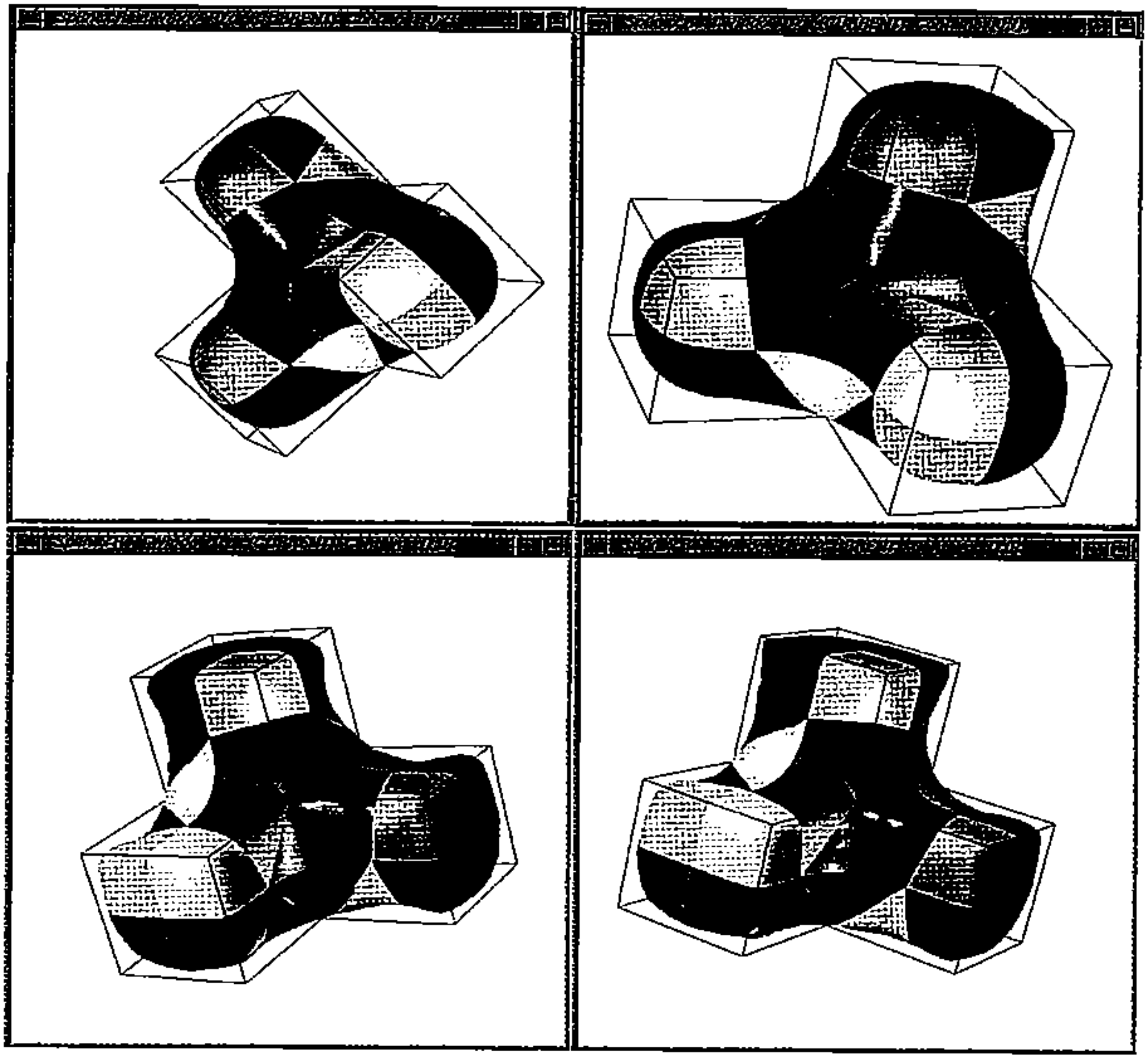


Figure 7.11: Shape Control of Smooth Approximations of a Polyhedron

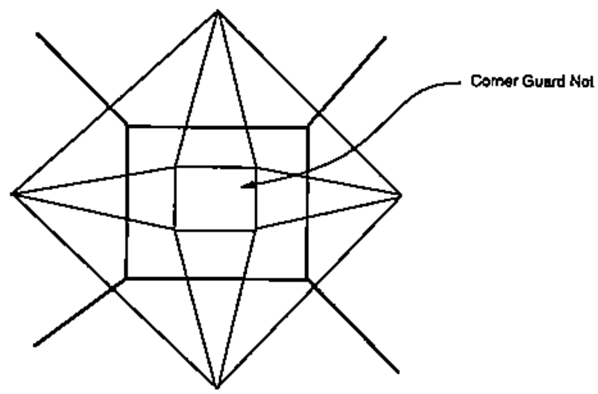


Figure 7.12: Interpolation of a Corner Guard Net

the vertex V, or the top of the face-tetrahedron, or as low as the bottom of the face-tetrahedron. For a nontrihedral corner that is to be smoothed in step (4) of Algorithm 1 and Algorithm 2, we adjust parameter α to decide the position of C_0 on the line segment $V\overline{V}$. Similarly, if C_0 is chosen close to V, only a little is cut off from the corner, while if C_0 is chosen close to $\overline{V}$, the corner is nearly almost all cut off. See Figure 7.11. In the top left figure, the weights are set approximating piecewise functional surface over the faces of the triangulation. In the top right figure, the upper leg of the surface is dragged to a corner by changing the weights of the tetrahedron on this corner. In the bottom left and bottom right figures, the whole surface is deformed in this manner in different scales.

7.2 Interpolating points

We introduce a corner guard net in the case of nonconvex polyhedral faces. The final surface interpolates the edges of the corner guard net. Actually, even in the case of convex face, we can make it a corner guard net that we would like the smooth surface to pass through certain lines, or even a region of the face. See Figure 7.12. In latter case, the corner guard "region" is covered by a planar polygon in the final piecewise smooth surface.

8 Implementation

We have presented algorithms for approximating a three dimensional polyhedron with C^1 cubic and C^2 quintic A-patches, respectively. These algorithms have been implemented in the SPLINEX and SHILP toolkits of our X-11 based distributed and collaborative geometric design environment SHASTRA [AB93]. See Figure 7.10 and Figure 7.11. We are using it for interactive free-form design. SHILP is an X-11 based, interactive solid modeling system and is used to create a simplicial (face triangulated) polyhedral model of the desired shape. This model could also be the triangulation of an arbitrary surface in three dimensions. This triangulation is C^1 smoothed by a client/server call to a SPLINEX computation using inter process communication. SPLINEX is a an X-11 based, interactive surface modeling toolkit for arbitrary algebraic surfaces (implicit or parametric) in BB form. It allows for the creation of simplex chains (as for example the simplicial hull of the triangulation) and the interactive change of control points and weights of the A-patches for shape control. SPLINEX also has the ability to distribute its rendering tasks (for the display of the individual A-patches) on a network of workstations, to achieve maximal display parallelism.

References

[AB93] V. Anupam and C. Bajaj. Collaborative Multimedia Scientific Design in SHASTRA. In *Proc. of the First ACM International Conference on Multimedia, ACM MULTIMEDIA 93*, pages 447–456. ACM Press, 1993.

[Baj92] C. Bajaj. Surface fitting with implicit algebraic surface patches. In H. Hagen, editor, *Topics in Surface Modeling*, pages 23 – 52. SIAM Publications, 1992.

[BBX94] C. Bajaj, F. Bernardini, and G. Xu. *Reconstruction of Surfaces and Surfaces-on-Surfaces from Unorganized Weighted Points.* Computer Science Technical Report, CS-93-001, Purdue University, 1994.

[BCX93] C. Bajaj, J. Chen, and G. Xu. *Interactive Modeling with A-Patches.* Computer Science Technical Report, CSD-TR-93-002, Purdue University, 1993.

[BI92a] C. Bajaj and I. Ihm. Algebraic surface design with Hermite interpolation. *ACM Transactions on Graphics*, 19(1):61–91, January 1992.

[BI92b] C. Bajaj and I. Ihm. C^1 Smoothing of Polyhedra with Implicit Algebraic Splines. *SIGGRAPH'92, Computer Graphics*, 26(2):79–88, 1992.

[BIW93] C. Bajaj, I. Ihm, and J. Warren. Higher-order interpolation and least-squares approximation using implicit algebraic surfaces. *ACM Trans. on Graphics*, 12(4):327–347, 1993.

[BX92] C. Bajaj and G. Xu. *A-Splines: Local Interpolation and Approximation using C^k-Continuous Piecewise Real Algebraic Curves.* Computer Science Technical Report, CAPO-92-44, Purdue University, 1992.

[Dah89] W. Dahmen. 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.

[DTS93] W. Dahmen and T-M. Thamm-Schaar. Cubicoids: modeling and visualization. *Computer Aided Geometric Design*, 10:93–108, 1993.

[Far90] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide.* Academic Press Inc., 1990.

[Guo91a] B. Guo. *Modeling Arbitrary Smooth Objects with Algebraic Surfaces.* PhD thesis, Computer Science, Cornell University, 1991.

[Guo91b] B. Guo. Surface generation using implicit cubics. In N.M. Patrikalakis, editor, *Scientific Visualizaton of Physical Phenomena*, pages 485–530. Springer-Verlag,Tokyo, 1991.

[Guo93] B. Guo. Non-splitting Macro Patches for Implicit Cubic Spline Surfaces. *Computer Graphics Forum*, 12(3):434 – 445, 1993.

[MW91] D. Moore and J. Warren. 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.

[Pet90] J. Peters. Local smooth surface interpolation: a classification. *Computer Aided Geometric Design*, 7:191–195, 1990.

[Pra87] V. Pratt. Direct least squares fitting of algebraic surfaces. *Computer Graphics*, 21(3):145–152, 1987.

[Sed90] T.W. Sederberg. Techniques for cubic algebraic surfaces, tutorial part ii. *IEEE Computer Graphics and Applications*, 10(5):12–21, Sept. 1990.

A Proofs Lemmas of Section 4

Lemma A.1 *(Lemma 4.1) Triangulation $\mathcal{T}$ is edge-convex.*

Proof. There are 3 kinds of edges in $\mathcal{T}$. (1) C_iC_j, edge between the centroids of two adjacent faces f_i and f_j. C_iC_j is convex since the face normal of f_i and f_j are the normals of C_i and C_j. (2) VC_i, edge between an unseparable vertex V and C_i, the centroid of a face f_i around vertex V. Edge VC_i is always convex as VC_i is always perpendicular to the normal at C_i, or the face normal of f_i. (3) C_0C_i, edge between C_0, the pseudo centroid of a separable vertex V and C_i, the centroid of a face f_i around vertex V. Without lost of generality, assume that $t_i \leq 0$. By the input assumption, $n_V n_{C_i} \geq 0$, namely n_V is pointing above face f_i. So C_0 is under face f_i. Therefore $\overrightarrow{C_iC_0}n_{C_i} \leq 0$. Let p_{C_i} be the projection of C_i on line $l_V = V + n_V t$. By construction, $|VC_0| < |Vp_{C_i}|$, hence $\overrightarrow{C_0C_i}n_{C_i} \leq 0$. Remember that $n_{C_i} = n_V$. So edge C_0C_i is convex. Therefore, every edge in $\mathcal{T}$ is convex. $\mathcal{T}$ is an edge-convex triangulation. This settles condition (1) of definition 3.6. $\diamond$

Lemma A.2 *In a polyhedron, for a corner V with normal n_V, if the inner product between n_V and the face normal n_{f_i} of any face f_i around V is positive, then the faces' projections on the tangent plane of V do not overlap each other.*

Proof. Assume that the lemma is not true. Then there must be two adjacent faces f_i and f_{i+1} whose projections overlap each other. One of them, say f_i, must "face downward", namely $n_{f_i} n_V < 0$, a contradiction. $\diamond$

Lemma A.3 *(Lemma 4.2) Every face-tetrahedron $[p_ip_jp_kq_l] \in S_f$ is tangent-containing.*

Proof. We divide the face-tetrahedra into following 5 groups. (1) $[C_1C_2C_3V]$, face-tetrahedra built in step (3) of Algorithm 1. (2) $[C_0T_0C_1C_2]$ is the face-tetrahedron built in step (5)(i) of Algorithm 1. (3) $[C_0C_1C_2T_1]$ is the face-tetrahedron built in step (5)(ii) of Algorithm 1. (4) $[C_0C_1C_2T_0]$ and $[C_0T_1C_1C_2]$, a pair of face-tetrahedra built in step (5)(iii) of Algorithm 1. (5) $[C_0C_1C_2T_0]$ and $[C_0T_1C_1C_2]$, a pair of face-tetrahedra built in step (5)(iv) of Algorithm 1 and $(n_{C_0}C_1)(n_{C_0}C_2) \geq 0$.

Let us look at the tangent plane of the centroids first. In case (1), by construction, $\overrightarrow{C_iV}n_{C_i} = 0$ for $i = 1, 2, 3$. So it is tangent-containing. In case (2), (4) the tangent planes of C_1 and C_2 are contained same as in case (1), $\overrightarrow{C_iT_0}n_{C_i} = 0$ for $i = 1, 2$, for T_0 is a point on edge e. In case (3), without lost of generality, assume that $(n_{C_0}C_1) \leq 0$, $(n_{C_0}C_2) \leq 0$, so that edge e is a ridge. As T_1 is the projection of $M \in e$ on C_0's tangent plane π_{C_0}, $\overrightarrow{MT_1}$ is parallel to n_V. Hence T_1 is above or on face f_1, or the inner product $n_{C_1}\overrightarrow{C_1T_1} > 0$. On the other hand, $n_{C_1}\overrightarrow{C_1C_2} \leq 0$ as e is a ridge, $n_{C_1}\overrightarrow{C_1C_0} \leq 0$ as C_0 is below face f_i. Hence at C_1, the tangent plane is contained. Symmetrically, tangent plane is also contained at C_2. Similarly, we prove that in case (5), the tangent planes at C_1 and C_2 are contained.

Now we consider the tangent plane of C_0 in cases (2)-(5). In case (2), π_{C_0} intersects $[T_0C_0C_1C_2]$ as the inner product $n_{C_0}\overrightarrow{C_0T_0}$ is of a different sign to that of the inner products $n_{C_0}\overrightarrow{C_0C_1}$ and

$n_{C_0}\overrightarrow{C_0C_2}$. Hence $[C_0C_1C_2T_0]$ is tangent-containing at C_0. In cases (3), (4) and (5), tangent containment is obvious as $n_{C_0}\overrightarrow{C_0T_1} = 0$.

From the above discussion, $\mathcal{S}_f$ is tangent-containing, which settles condition (4) of definition 3.6. $\diamond$

Lemma A.4 *(Lemma 4.3) The face-tetrahedra in $\mathcal{S}_f$ do not intersect each other.*

Proof. Now we show that adjacent face-tetrahedra do not intersect each other. Let $[C_1C_2C_3T_1]$ and $[C_1C_2C_4T_2]$ be two face-tetrahedra adjacent to each other. T_1 and T_2 can be either vertices on some edges or projections on the tangent planes of some vertices. First, assume C_1 and C_2 are the centroids of two adjacent faces incident at edge e, T_1 and T_2 are on the edge e(see Steps (3) (5)(i)(iii)(iv) of Algorithm 1), or the projection of M_e, the midpoint of e on the tangent plane of C_3 or C_4 (Steps (5)(ii)(iii)(iv) of Algorithm 1). Let σ be the plane passing through M_e and perpendicular to e. In any combination of the above cases, T_1 and T_2 are in different side of σ. Therefore the two tetrahedra do not intersect each other. Secondly, if C_1 is a pseudo centroid, C_2 must be a regular centroid. T_1 and T_2 are vertices on neighboring edges e_1 and e_2 or on e_1 and e_2's projections on π_{C_1}, the tangent plane of the pseudo centroid C_1. As the inner product $n_{C_0}n_{C_i} > 0$ for any surrounding face f_i , the projections of the surrounding faces and edges on π_{C_1} do not overlap each other (Lemma A.2). Let σ be the plane passing through C_1 and perpendicular to π_{C_1} and bisecting the angle $e_1C_1e_2$. In any combination, T_1 and T_2 are on different sides of σ. Therefore $\mathcal{S}_f$ is not self-intersecting, which satisfies condition (5)(i) of definition 3.6. $\diamond$

B Elimination of Coplanarity

In [BCX93], extra subdivision is needed if the adjacent triangular faces of the triangulation $\mathcal{T}$ are coplanar to each other. Otherwise, the bottom weights of the face-tetrahedra are all related to each other by the continuity constraints, so that the locality property of the weight-setting procedure is destroyed and also the single sheeted condition is jeopardized. In the scheme presented in this paper, however, as the polyhedron face is also constructed instead of being given, one can choose to construct a triangulation without any coplanar faces or only with some "trivial" coplanar cases.

Coplanarity of a triangulation $\mathcal{T}$ could be described by a coplanar graph, denoted as $\mathcal{G}$, which has an edge f_if_j if and only if face f_i and f_j are coplanar to each other.

The vertices of the triangulation are the corner guards of the faces of the original polyhedron $\mathcal{P}$ and pseudo centroids of the non-trihedral vertices of $\mathcal{P}$. Those corner guards and pseudo centroids are by no means in any unique positions related to the faces. One exception is the non-trihedral vertex that is unseparable from its corner guards, but such kind of vertices are never adjacent to each other in $\mathcal{T}$. Hence coplanarity between adjacent facts in $\mathcal{T}$ can be eliminated by displacement of the corner guards or the pseudo centroids.

In our implementation, instead of computing a corner guard, we compute a *corner guard circle,* in which every point is a corner guard. Corner guard circles of a face are connected into a *corner guard circle net.* We center a corner guard on each circle. But any of them can be moveed around within the circle without affecting any other part of the corner guard net. However, while being a complete solution to the coplanarity problem, displacement is not needed in the following two "trivial" coplanar cases.

(1) A face-tetrahedron is coplanar to only one other face-tetrahedron, namely the coplanar faces in $\mathcal{T}$ can be grouped into isolated pairs. In this case, the bottom weights of the two faces are

dependent to each other, but they are not dependent to other face-tetrahedra.

(2) A group of coplanar faces whose edges between themselves are all zero convex. The continuity constraint can be set among them by setting the bottom weights a_{1110}'s to zero. Let us refer to such a group as a *zero coplanar group*. A special case of this kind is a group like $VG_iG_{i+1} \in \mathcal{T}$, where vertex V is a non-trihedral vertex that is unseparable from the corner guards G_i around them, (see Algorithm 2 step (4)) and $VG_iG_{i+1} \in \mathcal{T}$, the triangular faces around V are on one common plane π_V. The normal at each corner guard G_i is vertical to the edge, $n_{G_i}(G_iV) = 0$. Now if we set the normal of V to be that of plane π, then all the edge VG_i are zero convex.

It can be shown that if $[p_ip_jp_k]$ is a face coplanar over a nonzero edge $[p_ip_j]$ to a face in a zero coplanar group, then p_k is *not* an unseparable vertex, and hence can be displaced to eliminate the coplanarity over $[p_ip_j]$.

Lemma B.1 *A zero coplanar group can be always removed from a coplanar graph $\mathcal{G}$ by displacing a vertex of the face coplanar to it.*

We sum up the above discussion by modifying Algorithm 2 to resolve the coplanarity problem without subdividing the triangulation. The significance of the corner guard circle in this modification is that after any displacement of a corner guard, the reconstruction of the simplicial hull is needed except when a displaced vertex is adjacent to an unseparable vertex $V \in \mathcal{P}$.

Algorithm 3 *INPUT:(same as in Algorithm 2)*

(1') Compute the corner guard circle net of each faces.

(2)(3')(3) (same as (2)(3')(3) in Algorithm 2)

(4') Let $l_V = V + n_V t$, be a straight line passes through vertex V in the direction of the normal n_V. Let C_{G_i} be guard circle on f_i of vertex V. Compute the projection of guard circle C_{G_i} on l_V, which is a line segment and denoted as $P^0_{G_i}P^1_{G_i}$, $P^0_{G_i} = V + n_V t^0_i$, $P^1_{G_i} = V + n_V t^1_i$. If $t^0_i, t^1_i > 0$ for all i or $t^0_i, t^1_i < 0$ for all i, then V is separable from the centroids C_i of the faces f_i around V, by a plane π parallel to the tangent plane at V, otherwise V is unseparable. In the case of a separable vertex V, let $\overline{V}$ be the projection closest to V, or the one with the smallest t^j_i in absolute value; $\overline{V} = V + n_V t^*$, where $t^* \in \{t^0_i, t^1_i\}$ and $|t^*| = min\{|t^0_i|, |t^1_i|\}$. In the case of an unseparable vertex V, let $\overline{V} = V$. Let C_0, denoted as a pseudo centroid, *be a point between V and $\overline{V}$. $C_0 = \alpha V + (1-\alpha)\overline{V}$ for some $\alpha \in (0,1)$. Let n_V be the normal at C_0. Go to (5).*

(5)(5') (same as in Algorithm 2.)

(5")(i) Construct planar graph $\mathcal{G}$.

(5")(ii) Look for zero coplanar groups. For each face-tetrahedron based on a face of an zero coplanar groups, set weights $a_{1110} = 0.0$.

(5")(iii) For each face $F = [p_ip_jp_k] \in \mathcal{T}$ that is coplanar to a zero coplanar group over edge $[p_ip_j]$, displace p_k within the corner guard circle as as to eliminate the coplanarity over edge $[p_ip_j]$.

(5")(iv) Break the rest of $\mathcal{G}$ into pairs by displace some corner guards within their corner guard circles.

(5")(v) For any unseparable vertex V adjacent to a displace guard G_i, reconstruct all the face-tetrahedra sharing $[VG_i]$, as well as the edge-tetrahedra adjacent to them.

(6) (same as in Algorithm 2.)

OUTPUT: Simplicial hull Σ without any coplanarity other than coplanar pairs or zero coplanar groups.

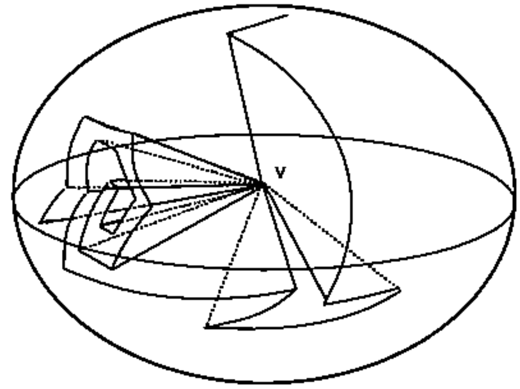


Figure C.13: A Complicated Corner

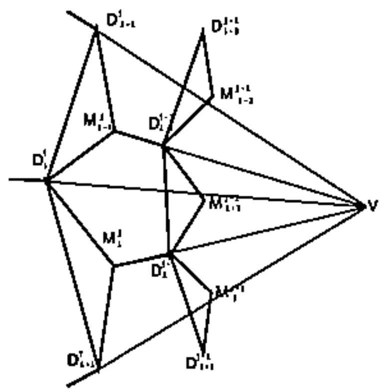


Figure C.14: Smooth an Unseparable Corner

C Transform Arbitrary a Simple Polyhedron into one that has only “above face” vertex normal

Intersect the faces of a vertex V with a sphere S centered at V. The intersection is a non-self-intersecting polygon $P = P_1P_2...P_n$ on the surface of S. For $i = 1, ..., n$, replace P_i by the midpoint of arc $P_iP_{(i+1)mod(n+1)}$. Repeat this until P's projection on some big circle C does not overlap itself. That means that if the plane of C is chosen as the tangent plane of V, then the projection of V's surrounding faces do not overlap each other and the inner product of the normal at V and any face normal around V is positive.

Map this process onto a vertex V. Let k be the number of iterations in the process. Initially, $j = 0$. (1) As shown in Figure C.14, on each edge E_i, take a point D_i^j so that $|D_i^jV| = 2k \cdot s$, where s is some preselected step size. Determine M_i^j so that $|M_i^jV| = (2k-1)s$ and $|D_i^jM_i^j| = |D_{i+1}^jM_i^j|$. Connect $D_i^jM_i^j$ and $D_{i+1}^jM_i^j$, VM_i^j and $M_i^jM_{i+1}^j$ delete D_i^jV.
(2) if k is 0 stop. Otherwise $k = k-1$ and $j = j+1$ go to (1).

The polygon $M_1^kM_2^k...M_n^k$ forms a corner V_u^n which is smoothable. The newly introduced vertices M_j^i are of vertex-degree 5 and D_j^i, D_{j+1}^i, D_j^{i+1} and M_j^i are coplanar, so that it can be shown that M_j^i are also smoothable.

D Determining Number 5 and 6 Weights of Quintic

It follows from (2.8) and (2.9) that

$$b_{1211}^{(i)} = \beta_1^{(i)} a_{1211}^{(i)} + \beta_2^{(i)} a_{0311}^{(i)} + \beta_3^{(i)} a_{0221}^{(i)} + \beta_4^{(i)} a_{0212}^{(i)} \tag{D.15}$$

$$\begin{aligned} b_{2210}^{(i)} &= \beta_1^{(i)}\beta_1^{(i)} a_{2210}^{(i)} + 2\beta_1^{(i)}\beta_2^{(i)} a_{1310}^{(i)} + 2\beta_1^{(i)}\beta_3^{(i)} a_{1220}^{(i)} + 2\beta_1^{(i)}\beta_4^{(i)} a_{1211}^{(i)} \\ &+ \beta_2^{(i)}\beta_2^{(i)} a_{0410}^{(i)} + 2\beta_2^{(i)}\beta_3^{(i)} a_{0320}^{(i)} + 2\beta_2^{(i)}\beta_4^{(i)} a_{0311}^{(i)} \\ &+ \beta_3^{(i)}\beta_3^{(i)} a_{0230}^{(i)} + 2\beta_3^{(i)}\beta_4^{(i)} a_{0221}^{(i)} + \beta_4^{(i)}\beta_4^{(i)} a_{0212}^{(i)} \end{aligned} \tag{D.16}$$

for $i = 1, 2$. (D.16) can be written briefly as

$$b_{2210}^{(i)} = 2\beta_1^{(i)}\beta_4^{(i)} a_{1211}^{(i)} + \beta_4^{(i)}\beta_4^{(i)} a_{0212}^{(i)} + \gamma \tag{D.17}$$

where γ is the known terms in (D.16). Since

$$b_{2210}^{(1)} = \mu_1 b_{1211}^{(1)} + \mu_2 b_{1211}^{(2)} \tag{D.18}$$

$$\mu_1^2 b_{0212}^{(1)} - \mu_1 b_{1211}^{(1)} = \mu_2^2 b_{0212}^{(2)} - \mu_2 b_{1211}^{(2)} \tag{D.19}$$

then by substituting (D.15) into (D.18) and (D.19) and then eliminating $b_{2210}^{(i)}$ from (D.17) and (D.18) we get three equations related to four unknowns which could be written as:

$$\begin{bmatrix} \beta_4^{(1)} - \mu_1 & -\mu_2 \\ -\mu_1 & \beta_4^{(2)} - \mu_2 \end{bmatrix} \begin{bmatrix} \beta_4^{(1)} & 0 \\ 0 & \beta_4^{(2)} \end{bmatrix} \begin{bmatrix} a_{0212}^{(1)} \\ a_{0212}^{(2)} \end{bmatrix} = -\begin{bmatrix} 2\beta_4^{(1)} - \mu_1 & -\mu_2 \\ -\mu_1 & 2\beta_4^{(2)} - \mu_2 \end{bmatrix} + \cdots \tag{D.20}$$

$$\begin{bmatrix} -\mu_1(\beta_4^{(1)} - \mu_1) & \mu_2(\beta_4^{(2)} - \mu_2) \end{bmatrix} \begin{bmatrix} a_{0212}^{(1)} \\ a_{0212}^{(2)} \end{bmatrix} - \begin{bmatrix} \mu_1\beta_1^{(1)}, -\mu_2\beta_1^{(2)} \end{bmatrix} \begin{bmatrix} a_{1211}^{(1)} \\ a_{1211}^{(2)} \end{bmatrix} = \cdots \tag{D.21}$$

where $\cdots$ are known terms. Since the coefficient matrix of (D.20) is nonsingular, by solving $[a_{0212}^{(1)}\ a_{0212}^{(2)}]^T$ from (D.20) and then substituting it into (D.21), we get one equation relating to the unknowns $a_{1211}^{(1)}$, $a_{1211}^{(2)}$. Let the equation be in the form

$$\phi a_{1211}^{(1)} + \psi a_{1211}^{(2)} = \omega \tag{D.22}$$

Therefore, these unknowns form a closed chain around the vertex p_2. But one should note that the chain will go to the other side of the triangles if the edges change their convexity(from positive convex to negative convex or from negative convex to positive convex). For example, if the edge $[p_2p_3]$ is positive convex, while the edge $[p_1p_2]$ is negative convex, then the chain is changed by adding one more equation related to the two unknowns $a_{1211}^{(1)}$ and $c_{1211}^{(1)}$ from C^1 condition

$$c_{1211}^{(1)} = \alpha_1 a_{2210}^{(1)} + \alpha_2 a_{1310}^{(1)} + \alpha_3 a_{1220}^{(1)} + \alpha_4 a_{1211}^{(1)}$$

where

$$q_4 = \alpha_1 p_1 + \alpha_2 p_2 + \alpha_3 p_3 + \alpha_4 p_4$$

Again, this equation is in the same form as (D.22). The coefficient matrix of all these equations related to the vertex p_2 is in the form of

$$A = \begin{bmatrix} \phi_1 & \psi_1 & & \\ & \phi_2 & \psi_2 & \\ & & \ddots & \\ \psi_{k+r} & & & \phi_{k+r} \end{bmatrix}$$

whose determinant is $\prod_{i=1}^{k+r} \phi_i - (-1)^{k+r} \prod_{i=1}^{k+r} \psi_i$. This matrix is nonsingular in general if the points given are in the general position. Hence the system can be solved.

In this specified case, $a^{(1)}_{1202}$ and $a^{(1)}_{2102}$ do not involve in any equation, since there is no neighbor tetrahedron. These two weights are defined by a C^2 condition crossing the face $[p_1p_2p_3]$.

E Determining Number 9 Weights of the Quintic Scheme

For $i = 1, 2$,

$$b^{(i)}_{1202} = \beta^{(i)}_1 a^{(i)}_{1202} + \beta^{(i)}_2 a^{(i)}_{0302} + \beta^{(i)}_3 a^{(i)}_{0212} + \beta^{(i)}_4 a^{(i)}_{0203} \tag{E.23}$$

$$\begin{aligned} b^{(i)}_{2201} &= \beta^{(i)}_1\beta^{(i)}_1 a^{(i)}_{2201} + 2\beta^{(i)}_1\beta^{(i)}_2 a^{(i)}_{1301} + 2\beta^{(i)}_1\beta^{(i)}_3 a^{(i)}_{1211} + 2\beta^{(i)}_1\beta^{(i)}_4 a^{(i)}_{1202} \\ &+ \beta^{(i)}_2\beta^{(i)}_2 a^{(i)}_{0401} + 2\beta^{(i)}_2\beta^{(i)}_3 a^{(i)}_{0311} + 2\beta^{(i)}_2\beta^{(i)}_4 a^{(i)}_{0302} \\ &+ \beta^{(i)}_3\beta^{(i)}_3 a^{(i)}_{0221} + 2\beta^{(i)}_3\beta^{(i)}_4 a^{(i)}_{0212} + \beta^{(i)}_4\beta^{(i)}_4 a^{(i)}_{0203} \end{aligned} \tag{E.24}$$

and

$$b^{(1)}_{3200} = \mu_1 b^{(1)}_{2201} + \mu_2 b^{(2)}_{2201} \tag{E.25}$$

$$\mu_1^2 b^{(1)}_{1202} - \mu_1 b^{(1)}_{2201} = \mu_2^2 b^{(2)}_{1202} - \mu_2 b^{(2)}_{2201} \tag{E.26}$$

Substitute (E.23) and (E.24) into (E.26), we have

$$\mu_1\beta^{(1)}_4(\mu_1 - \beta^{(1)}_4) b^{(1)}_{0203} - \mu_2\beta^{(2)}_4(\mu_2 - \beta^{(2)}_4) b^{(2)}_{0203} = \cdots$$

This is a system that is in the same form as (D.22) and a system like (D.22) needs to be solved. However, if the surrounding tetrahedra at the same side at p_2 are not closed, the matrix A is in the form of

$$A = \begin{bmatrix} \phi_1 & \psi_1 & & \\ & \ddots & \ddots & \\ & & \phi_k & \psi_k \end{bmatrix}$$

By chosing one unknown, say the l-th to be a free parameter, A can be written as $A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}$ with

$$A_1 = \begin{bmatrix} \phi_1 & \psi_1 & & \\ & \ddots & \ddots & \\ & & \phi_{l-1} & \psi_{l-1} \\ & & & \phi_l \end{bmatrix}, \quad A_2 = \begin{bmatrix} \psi_{l+1} & & & \\ \phi_{l+2} & \psi_{l+2} & & \\ & \ddots & \ddots & \\ & & \phi_k & \psi_k \end{bmatrix}$$

Hence the system of equations breakup into two smaller sub-systems. Each of them can be solved separately.