UNIVERSITY OF WISCONSIN-MADISON
CENTER FOR THE MATHEMATICAL SCIENCES

LOCAL PIECEWISE CUBIC C^1 SURFACE INTERPOLANTS FOR RECTANGULAR AND TRIANGULAR TESSELLATIONS

by
Jörg Peters

Technical Summary Report # 89-10
September 1988

Abstract

Consider points in 3-space equipped with normals and an ordered list of neighbors such that the corresponding piecewise [bi]linear interpolant to the data consists of 3- or 4-sided facets. This paper describes an algorithm which, for appropriate data, constructs a piecewise cubic C^1 surface interpolant with one patch per 3- or 4-sided facet. The interpolating surface depends continuously on local data and its construction only requires the solution of linear systems of equations.

AMS (MOS) Subject Classifications: 41A15, 41A10, 41A05
Key words: Bernstein-Bézierform, C^1 surface, interpolation, cusping, shape control

This research was supported by NSF DMS-8701275

1. Introduction

Consider points in 3-space equipped with normals and an ordered list of neighbors such that the corresponding piecewise [bi]linear interpolant to the data consists of 3- or 4-sided facets. This paper describes an algorithm which, under certain conditions on the data, constructs a piecewise cubic surface interpolant whose normal changes continuously. The interpolating surface is well behaved as a function of the data: it depends continuously on the data and only on local data. The construction is efficient: the polynomial pieces are of low degree (cubic, resp. bicubic) and there is exactly one polynomial piece per facet. Only linear systems of equations have to be solved. In fact, the scheme is linear in the normal data and linear in the data points.

The algorithm has two problems. The input data at adjacent points of the tessellation must allow for a linearly changing normal direction along the connecting boundary. This may be achieved by either adding points or choosing the normals at the data points appropriately. But, at present, no satisfactory strategy for either type of improvement is known. Cusping is another problem of the algorithm. Although this paper derives explicit (necessary and sufficient) conditions on the data under which the surface is cusp-free, the conditions are complicated. Again the choice of normals plays a crucial role.

We classify the algorithm as "f[3,4]p1d3c1" since it associates each 3- or 4-sided facet with one polynomial patch and constructs, for appropriate data, a piecewise cubic C^1 surface.

We use the Bernstein-Bézier representation for the polynomial pieces, because this form gives easy access to value and derivative information along the *edges* of a patch. See [Farin '86] or [de Boor '87] for an overview on polynomials in the Bernstein-Bézier form. Determining the surface thus corresponds to computing the Bézier coefficients each of which is a 3-vector. Since we look at the product of polynomials in the BB-form, we find it advantageous to use the abbreviation

$$\underbrace{\{\alpha,\ldots,\beta,\ldots,\gamma\}}_{l+1 \text{ terms}}$$

to stand for the polynomial

$$u \mapsto \alpha(1-u)^l + \ldots + \beta(1-u)^{l-j}u^j + \ldots + \gamma u^l.$$

In Section 2, we outline the character of the problem in more detail and introduce the idea behind f[3,4]p1d3c1. Section 3 presents and analyzes the algorithm. Section 4 shows what conditions on the data guarantee cusp-free surfaces. Section 5 introduces curvature weights as an important improvement over the basic algorithm and Section 6 gives examples.

2. Dissecting the problem

First, we resolve some minor issues. With each facet we associate one Bézier patch such that the corner coefficients match the respective data points. Then we ensure a C^0 match between the patches: any two patches sharing two data points must share a common boundary polynomial. We may then use the parametrization of any adjacent pair of boundary curves of a patch to parametrize that patch.

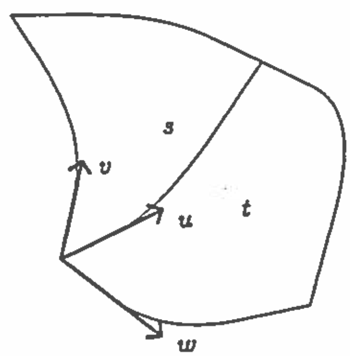

Figure 2: Abutting patches.

Consider a patch s parametrized by u and v and its neighbor patch t parametrized by u and w. We use subscripts to denote partial derivatives, e.g. $s_u := \frac{\partial}{\partial u} s(u,v)$. Our goal is to construct a surface that is C^1 across the common boundary, i.e. a surface whose normal is uniquely defined at every point of the boundary (and changes continuously across):

$$s_v \times s_u \neq 0, \tag{1}$$

$$\frac{s_v \times s_u}{\|s_v \times s_u\|} = \frac{s_u \times t_w}{\|s_u \times t_w\|}, \tag{2}$$

at each point of the boundary. Note that we were careful with the order in the vector products.

Thus our task is, in general,

- nonlinear,
- global,
- involves inequalities as well as equality constraints.

We can rephrase the conditions to dispense with the normalization.

Lemma 2.1. *A match between two polynomial patches $s(u,v)$ and $t(u,w)$ across a common boundary parametrized by u is C^1 if and only if*

$$((s_v(u,0) \times s_u(u,0)) \times s_u(u,0))t_w(u,0) > 0 \tag{3}$$

and

$$(s_v(u,0) \times s_u(u,0))t_w(u,0) = 0. \tag{4}$$

Proof: We will use the identities $(a \times b)(c \times d) = \det((a \times b), c, d) = ((a \times b) \times c)d$ throughout. First we show that (1) and (2) imply (3) and (4). To see that (4) holds we need only multiply (2) by $t_w \|s_v \times s_u\|$. The inequality (3) is implied by $s_v \times s_u \neq 0$ together with (2) since $0 < (s_v \times s_u)(s_u \times t_w) = ((s_v \times s_u) \times s_u)t_w$.

To show the converse, we observe that $0 < ((s_v \times s_u) \times s_u)t_w = \det((s_v \times s_u), s_u, t_w) = \det(s_v, s_u, (s_u \times t_w))$. Thus $s_v \times s_u \neq 0$ and $s_u \times t_w \neq 0$. Furthermore, (4) shows that $t_w \perp (s_v \times s_u)$. Hence $(s_u \times t_w)$ and $(s_v \times s_u)$ are collinear. Finally, since $0 < ((s_v \times s_u) \times s_u)t_w = (s_v \times s_u)(s_u \times t_w)$, we may conclude that the normalized vectors are the same. ♠

The inequality (3) is a regularity condition. It implies the existence of a unique normal on each patch. Equation (4) expresses perpendicularity of the normal of patch s to the tangent plane of t at each point of the boundary. Together the two conditions imply uniqueness of the normal across the boundary. Note that the order of the terms in (3) is important. The direction of the inequality distinguishes the C^1 match from a cusping match.

We observe that the C^1 conditions, (3) and (4), link *interior* Bézier coefficients both along the patch boundaries and around the data points. The resulting system of equations is therefore indeed not block diagonal but globally connected.

The central idea, which allows us to overcome these difficulties, is to impose additional conditions where the polynomial pieces meet and thus dissect some of the linking equations. In particular, we resolve all nonlinearity and global connectedness arising from (4) by prescribing a normal direction n along the boundaries. A similar idea, motivating this approach, was proposed in [Sabin68]. With u the parameter along the common boundary, the equality constraints (4) for patch s now read:

$$n(u)s_u(u,0) = 0, \tag{5}$$
$$n(u)s_v(u,0) = 0. \tag{6}$$

To be consistent, the normal directions which we choose must agree at the data points. We use N^i to denote the corresponding normalized vector at the data point P^i, e.g. N^i is the normal of the constructed surface at P^i. Since we want n to fit into the polynomial frame work, we choose n as a (rational) polynomial. To minimize the number of conditions in (5) and (6) the degree of the polynomial should be as low as possible. We find the minimal degree of n by looking at two adjacent points P^i and P^j and their difference,

$$\delta^{ij} := P^i - P^j.$$

If $N^i\delta^{ij} \leq 0$ (≥ 0) we say the **P^j neighborhood of P^i is concave (convex).** If the shapes of the neighborhoods at P^i and P^j agree, that is if

$$\alpha^{ij} := \frac{N^i\delta^{ij}}{N^j\delta^{ji}} \geq 0, \tag{7}$$

then $s(u,0)$ need not have a point of inflection in $[0,1]$, and we can choose n to be linear. If $\alpha^{ij} < 0$, however, n must be at least quadratic (due to the inflection). We are now ready to derive the algorithm.

3. The algorithm

In the following we consider only data that allow for a linear normal direction along the boundaries. That is, we make the

Assumption 3.1. $\alpha^{ij} \geq 0$.

3.0. The input

The input to the algorithm consists minimally of the data points and an ordered list of neighbors for each data point such that the resulting surface tesselation consists of pieces with 3 or 4 vertices. If the normal at the data points is part of the input, it must satisfy the restrictions of Assumption 3.1 as well as of Convention 3.2 below. If the normal is not prescribed, it is derived from the data, with the orientation given by

Convention 3.2. *In the neighbor list of any point P the neighbors are ordered clockwise when viewed from "above", i.e. when viewed in the direction opposite to the normal direction at P.*

Since only the relative order of the neighbors is of importance, *all index computations for the neighbor list are modulo the number of neighbors of the data point.* We will use the letter k for the number of neighbors of a data point. To obtain the patch structure from the input data, let q_p be the index of point p in the neighborlist of q (and p_q the index of q in the list of p). Further, we use the notation *structure.component*. A face f, for example, has components *type*, *vertex*, etc., with *vertex* an array of pointers to data points. Similarly the data points (p and q below) have an array component *patches* with pointers to faces. (The data structures reflect the two stages of the algorithm as will be clear at the end of Section 3). The following algorithm associates patches and vertices:

```
for each data point i
  for each neighbor j of i
    if j has not yet been visited from i
      f := new face;
      p := i;  q := j;
      bdy := 0;
      do /*walk counter-clockwise along the patch boundaries*/
        f.vertex[bdy] := p;
        p.patch[p_q − 1] := f;
        p := q;
        q := neighbor (q_p + 1) of q;  /*i.e. q := q.nbr[q_p + 1] */
        bdy := bdy + 1;
      until p == i;
      if bdy ∉ {3,4} error; /*not a 3- or 4-sided facet*/
      f.type := bdy;
```

3.1 Derivation of the equality constraints

In the following we restrict our attention to cubic polynomial patches. Recall the abbreviation $\{\alpha, \ldots, \binom{l}{j}\beta, \ldots, \gamma\}$ for the polynomial $u \mapsto \alpha(1-u)^l + \ldots + \binom{l}{j}\beta(1-u)^{l-j}u^j + \ldots + \gamma u^l$. In particular, the quadratic polynomial $\frac{1}{3}s_u(u,0) =: d^0(1-u)^2 + 2d(1-u)u + d^1u^2$ is abbreviated $\{d^0, 2d, d^1\}$ and the linear interpolant, n, to the normal directions at $u = 0$ and $u = 1$ is chosen to be

$$n(u) := \{n^0, n^1\} := \{N^0, N^1\}.$$

We ignore for now the inequality constraints (3) associated with the general problem. That is, the scheme need not be able to distinguish between a C^1 and a cusping match. We are then left to enforce conditions (5) and (6) along the edges. Equation (5) is satisfied for a particular edge if all coefficients of $\{d^0, 2d, d^1\}\{n^0, n^1\}$ are zero, i.e. iff

$$\{n^0d^0, 2n^0d + n^1d^0, 2n^1d + n^0d^1, n^1d^1\} = 0. \tag{8}$$

Thus we have a list of conditions for the boundary coefficients.

Similarly, with $\{e^0, 3E^0, 3E^1, e^1\} := \frac{1}{3}s_v(u,0)$ for a bicubic patch, equation (6) requires in general that all coefficients of $\{e^0, 3E^0, 3E^1, e^1\}\{n^0, n^1\}$ be zero, i.e.

$$\{n^0e^0, 3n^0E^0 + n^1e^0, 3n^0E^1 + 3n^1E^0, 3n^1E^1 + n^0e^1, n^1e^1\} = 0. \tag{9}$$

This yields a second list of conditions. We refer to the individual equations of each list by equation number and subscript, e.g.

$$(9_1) \qquad 3n^0E^0 + n^1e^0 = 0.$$

Note that enforcing (8_0) and (8_3) for *all* edges of the tessellation also enforces (9_0) and (9_4).

In the case of a bicubic patch it is now tempting to proceed as follows. First, enforce (8_0) through (8_3) for all edges. Since the two interior boundary coefficients offer six degrees of freedom, we can specify the tangent direction of the boundary curve and satisfy conditions (8_1) and (8_2) by adjusting the lengths of the tangent vectors.

We then choose the 4 interior Bézier coefficients so that equations (9_1) through (9_3) hold for each of the four sides of the patch. Unfortunately, the corresponding 12×12 system has rank at most 8 in this setting and in general no solution, as we shall now show. Let the two adjacent edges that connect the data point P with the points P^i and P^j be parametrized by u and v respectively. Then the system of equations is contradictory unless

$$n^i s_v = n^j s_u \quad \text{at } P. \tag{10}$$

This follows from equations (9_1) and (8_0) for the edge parametrized by u, and from (9_3) and (8_3) for the edge parametrized by v and the smoothness (twice continuous differentiability) of the patch parametrized by u and v:

$$\begin{aligned} n^i s_v &= n_u s_v = -ns_{vu} \\ &= -ns_{uv} = n_v s_u = n^j s_u \end{aligned} \quad \text{at } P.$$

Note that condition (10) depends entirely on the *boundary* coefficients of the patch.

Thus we have to add (10) to the list of equations for the boundary coefficients. However, once (10) is enforced, one equation of each pair of equations (9_1) and (9_3) is redundant and we may drop it from the second list.

The same argument holds in the case of a triangular patch. That is, we need to enforce (10) and (8) for the boundary coefficients. To obtain

$$\{e^0, 2e, e^1\}\{n^0, n^1\} = \{n^0 e^0, 2n^0 e + n^1 e^0, 2n^1 e + n^0 e^1, n^1 e^1\} = 0. \tag{11}$$

It then suffices to enforce (11_1).

3.2. Computing the boundary polynomials

The equations of type (10) glue the boundary curves together at the data points. Fortunately, we can reduce the system again to a local system by observing the following. Once (8_0) and (8_3) hold, (8_1) and (8_2) along the boundary connecting P^i to P^j are equivalent to

$$\begin{aligned} n^1 d^0 &= \frac{2}{3}\delta^{ij}(n^0 + 2n^1), \\ n^1 d^1 &= \frac{2}{3}\delta^{ji}(n^1 + 2n^0). \end{aligned}$$

This separation leaves us with a *linear system of equations for each data point* with 3 equations for each neighbor. The first of these equations corresponds to (8_0) and forces the tangent coefficients d^i to lie in the tangent plane. The second equation, corresponding to (11), forces the boundary curve to be perpendicular to the prescribed normal direction. The third equation, corresponding to (10), enforces the smoothness condition at the data point.

We display the system at some data point P with normal N and neighbors P^1 through P^k. All boundary curves are parametrized so as to start at P. In particular, n corresponds to $n(0) = N$ and n^i to the normal at P^i for $i \in \{1, \ldots, k\}$. The tangent vector at P of the boundary curve connecting P to P^i is denoted by d^i. 3×3 blocks of the form

$$\begin{pmatrix} n \\ n^i \\ n^{i+1} \end{pmatrix}$$

line the diagonal of the matrix which has almost, but not quite, circulant structure:

$$\begin{pmatrix} n & 0 & 0 & \dots & 0 \\ n^1 & 0 & 0 & \dots & 0 \\ n^2 & -n^1 & 0 & \dots & 0 \\ 0 & n & 0 & \dots & 0 \\ 0 & n^2 & 0 & \dots & 0 \\ 0 & n^3 & -n^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -n^{k-1} \\ 0 & 0 & 0 & \dots & n \\ 0 & 0 & 0 & \dots & n^k \\ -n^k & 0 & 0 & \dots & n^1 \end{pmatrix} \begin{pmatrix} d^1 \\ \vdots \\ d^k \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{2}{3}\delta^1(n+2n^1) \\ 0 \\ 0 \\ \frac{2}{3}\delta^2(n+2n^2) \\ 0 \\ \vdots \\ 0 \\ 0 \\ \frac{2}{3}\delta^k(n+2n^k) \\ 0 \end{pmatrix}. \tag{12}$$

We will refer to the system of type (12) at the data point P^l as (12^l).

Under what conditions is (12) solvable? To simplify the analysis, we transform the coordinate system rigidly, preserving orientation and angle, so that P is mapped into the origin, and n into (0, 0, 1). By (8_0), the third component of d^i in the new coordinate system must be zero, i.e. the first two components of d^i determine d^i uniquely. We will therefore use d^i also to denote the 2-vector solving the restriction of (12) to the tangent plane, i.e. $(12')$ below. Also by (8_0), we may restrict attention to the first two components of the projections of the weighted normals n^i into the tangent plane at P. We denote the 2-vector corresponding to these first two components by m^i.

Thus we arrive at the equivalent "reduced" $2k \times 2k$ system

$$\begin{pmatrix} m^1 & 0 & 0 & \dots & 0 \\ m^2 & -m^1 & 0 & \dots & 0 \\ 0 & m^2 & 0 & \dots & 0 \\ 0 & m^3 & -m^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -m^{k-1} \\ 0 & 0 & 0 & \dots & m^k \\ -m^k & 0 & 0 & \dots & m^1 \end{pmatrix} \begin{pmatrix} d^1 \\ \vdots \\ d^k \end{pmatrix} = \begin{pmatrix} \frac{2}{3}\delta_1(n+2n^1) \\ 0 \\ \frac{2}{3}\delta_2(n+2n^2) \\ 0 \\ \vdots \\ 0 \\ \frac{2}{3}\delta_k(n+2n^k) \\ 0 \end{pmatrix}, \tag{12'}$$

We say that a **triple of normals is edge-adjacent** if the corresponding data points are connected by two boundary curves. With this definition, we first check solvability under

Assumption 3.2. *Any 3 edge-adjacent normals are linearly independent.*

Assumption 3.2 implies that there exist unique scalars $\gamma^i \neq 0$ and β^i such that

$$m^{i-1} = \beta^i m^i + \gamma^i m^{i+1} \quad \text{for } i \in \{1, \dots, k\}. \tag{13}$$

In terms of

$$B^i := \det(m^{i-1} m^{i+1}) \text{ and } D^i := \det(m^i m^{i+1}) \neq 0$$

these scalars can be determined as

$$\gamma^i = -\frac{D^{i-1}}{D^i} \quad \text{and} \quad \beta^i = \frac{B^i}{D^i}. \tag{14}$$

We now use equation (13) to eliminate the only term below the diagonal band of invertible 2×2 matrices in (12′), i.e. we block eliminate. Since, in general, a new term appears in the bottom row of the column to the right, we apply (13) $k-1$ times. At each step the right hand side of the last equation accumulates an additional term of the form $\beta^j \gamma^1 \cdots \gamma^{j-1} r^j$, and the entry in the bottom row of the $j+1^{st}$ column is transformed into $-\gamma^1 \cdots \gamma^j m^j$. The final entry on the bottom of the right hand side is

$$s := \beta^1 r^1 + \beta^2 \gamma^1 r^2 + \ldots + \beta^{k-1} \gamma^1 \cdots \gamma^{k-2} r^{k-1},$$

where

$$r^i := \frac{2}{3} \delta_i (n + 2n^i).$$

The final entry in the bottom row of the $2k^{th}$ column is

$$\tilde{m} := -\gamma^1 \cdots \gamma^{k-1} m^{k-1} + m^1.$$

Since $\gamma^1 \cdots \gamma^i = (-)^i D^k / D^i$, we can write more shortly

$$\beta^i \gamma^1 \cdots \gamma^{i-1} = (-)^{i-1} D^k \frac{B^i}{D^{i-1} D^i} =: (-)^{i-1} D^k c^i.$$

The c^i play an important role in the further analysis. They can be interpreted as

$$c^i = \frac{\mathrm{Vol}^{i-1\,i+1}}{\mathrm{Vol}^{i-1\,i}\,\mathrm{Vol}^{i\,i+1}},$$

with $\mathrm{Vol}^{ij} := \det(N, n^i, n^j)$. The expressions for s and $\tilde{m}$ simplify to:

$$s = -D^k \sum_{j=1}^{k-1} (-)^j c^j r^j \quad \text{and} \quad \tilde{m} = \frac{(-)^{k-1}}{\gamma^k} m^{k-1} + m^1. \tag{15}$$

We now have the block upper triangular system

$$(12'') \qquad \begin{pmatrix} m^1 & 0 & 0 & \ldots & 0 \\ m^2 & -m^1 & 0 & \ldots & 0 \\ 0 & m^2 & 0 & \ldots & 0 \\ 0 & m^3 & -m^2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & -m^{k-1} \\ 0 & 0 & 0 & \ldots & m^k \\ 0 & 0 & 0 & \ldots & \tilde{m} \end{pmatrix} \begin{pmatrix} d^1 \\ \vdots \\ d^k \end{pmatrix} = \begin{pmatrix} r^1 \\ 0 \\ r^2 \\ 0 \\ \vdots \\ 0 \\ r^k \\ s \end{pmatrix},$$

which allows us to show the following theorem.

Theorem 3.1. *Under Assumption 3.2, (12) is of full rank if and only if P has an odd number of neighbors.*

Proof: Since (12) is invertible if and only if (12′) is, and since the transformation from (12′) to (12″) is rank preserving, we can concentrate on the bottom 2×2 subsystem,

$$\begin{pmatrix} m^k \\ \tilde{m} \end{pmatrix} (d^k) = \begin{pmatrix} r^k \\ s \end{pmatrix} \tag{16}$$

of (12″). By Assumption 3.2, the matrix in (12″) has full rank if and only if

$$M_2 := \begin{pmatrix} m^k \\ \tilde{m} \end{pmatrix}$$

has full rank. M_2 is singular if and only if

$$\frac{(-)^{k-1}}{\gamma^k} m^{k-1} + m^1 = \nu m^k \tag{17}$$

or, equivalently,

$$m^{k-1} = (-)^{k-1} \gamma^k \nu m^k + (-)^k \gamma^k m^1. \tag{18}$$

Comparing this to (13) with $i = k$, we see that (12″) and thus (12) is of full rank if and only if

$$(-)^k \gamma^k \neq \gamma^k \tag{19}$$

or, equivalently, k is odd. ♠

If P has an *even* number of neighbors, we focus on the right hand side of (16).

Theorem 3.2. *If Assumption 3.2 holds and P has an even number of neighbors, then (12) has (a 1-dimensional set of) solutions if and only if*

$$S^k := -\sum_{j=1}^{k} (-)^j c^j r^j = 0. \tag{20}$$

Proof: Since k is even, (18) takes the form

$$m^{k-1} = -\nu \gamma^k m^k + \gamma^k m^1.$$

Thus $\nu = -\beta^k/\gamma^k$. Applying the last step of Gaussian elimination (with multiplier $-\beta^k/\gamma^k$) to (16), we find that the left hand side of the last row is zero:

$$\tilde{m} + \frac{\beta^k}{\gamma^k} m^k = \frac{1}{\gamma^k}(-m^{k-1} + \gamma^k m^1 + \beta^k m^k) = 0.$$

Hence the right hand side must be zero, too, to have solvability. The right hand side equals

$$s + \frac{\beta^k}{\gamma^k} r^k = -D^k \sum_{j=1}^{k-1} (-)^j c^j - \frac{B^k D^k}{D^k D^{k-1}} r^k = D^k S^k.$$

Since $D^k \neq 0$, this completes the proof. ♠

Remark: Since k is even, S^k does not change if we choose a different neighbor to be first in the neighbor list. This is not true when k is odd!

As a special case, we have

Corollary 3.2. *If $\beta^i = 0$ for all $i \in \{1, \ldots, k\}$, k even, then (12) has a 1-dimensional set of solutions.*

Proof: $c^i = \beta^i(r^i/D^{i-1})$. ♠

For example, if the projected normals of all odd-numbered neighbors are pairwise linearly dependent and the projected normals of all even numbered neighbors are pairwise linearly dependent, then all β^i are 0. In particular, consider the

Example: If the data are six equally distributed points on a sphere, i.e. such that the C^0 interpolant is an octahedron, then each of the 8 systems (12) has a 1-dimensional set of solutions.

We now dispense with Assumption 3.2 and look at *dependent* triples $\{N^i, N^j, N^k\}$ of edge-adjacent normals. We discern two types of dependencies:

(a) collinearity of a pair $\{N^\alpha, N^\beta\} \subset \{N^i, N^j, N^k\}$ of normals connected by an edge and

(b) coplanarity of the triple $\{N^i, N^j, N^k\}$ of normals without collinearity in any of its pairs.

We say that a system (12^l) is **associated with the triple** $\{N^i, N^j, N^k\}$ **of dependent edge-adjacent normals** if either the full triple of normals or two collinear normals of that triple appear in the system.

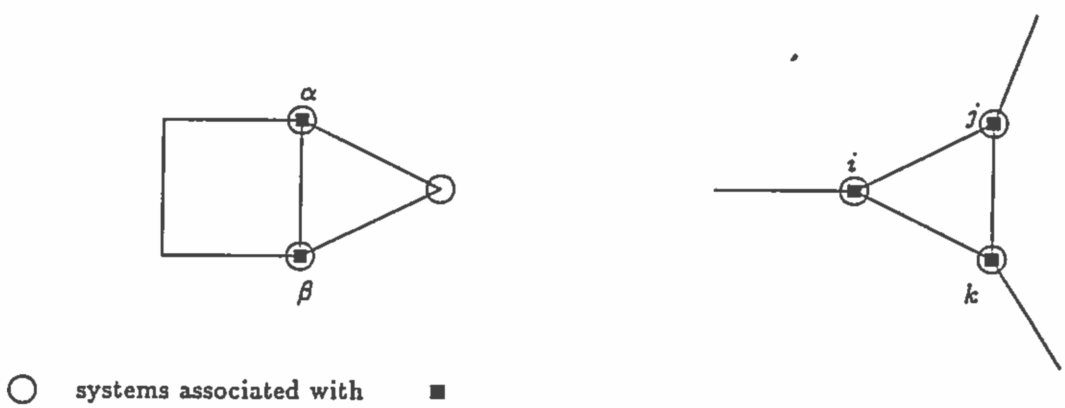

Figure 3.a: Associated equations.

Thus, if the dependency is of type (b), then exactly (12^i), (12^j), and (12^k) are associated with $\{N^i, N^j, N^k\}$. If the dependency is of type (a), then (12^α) and (12^β) are associated with $\{N^i, N^j, N^k\}$ as is any (12^l) such that both P^α and P^β are neighbors of P^l.

With this definition, we can capture all cases not treated in Theorem 3.1 and Theorem 3.2.

Theorem 3.3. *Let $\{N^i, N^j, N^k\}$ be a triple of dependent edge-adjacent normals. Let m^i and m^j be the projections of n^i and n^j into the tangent plane at P^k, and let $\eta^{ij} \neq 0$ be defined by $m^i =: \eta^{ij} m^j$. Then the systems associated with $\{N^i, N^j, N^k\}$ are solvable if* *and only if*

$$(\eta^{ij})^2\delta^{ki}(n^k+2n^i)=\delta^{kj}(n^k+2n^j) \tag{21}$$

and, for any pair $\{N^\alpha, N^\beta\}$, $\{\alpha,\beta\}\subset\{i,j,k\}$, *of edge-adjacent collinear normals,*

$$\delta^{\alpha\beta}n^\alpha=0. \tag{22}$$

Proof: First, we look at dependencies of type (a). Let N^α and N^β be linearly dependent normals. Then the projection of n^j into the tangent plane at P^i is the zero vector and the corresponding equation of type (11) in $(12'^i)$ is equivalent to (22).

To establish (21), consider $(12'^k)$. We may assume that P^i is the first and P^j is the second neighbor of P^k. Then (21) is implied by the first three equations of $(12'^k)$.

Conversely, once (21) holds, the second equation of $(12'^k)$ is redundant. Thus we may drop the second equation and move the last equation of $(12'^k)$ into its place to obtain a matrix of size $2k-1\times 2k$ with no terms below the diagonal band of $(k-1)$ 2×2 matrices or the single entry on the diagonal in the $2k-1^{st}$ row. We can now backsolve by virtue of either (22), (21) or the invertibility of the 2×2 matrices on the diagonal. ♠

A typical case of the situation analyzed in Theorem 3.3 is captured in the following

Example: Consider patches on a cylinder without lids. On such a cylinder any 3 edge-adjacent normals lie in a plane. If we choose, for example, a triangular patch such that two data points lie on one of the generating circles and the third is at equal distance from the two points on a different generating circle, then (21) holds.

Remark: If neither of the above conditions on the data are met, the algorithm has to either perturb normals or request additional information.

3.3. Computing the center coefficients

Having ensured that (10) and (8_0) hold at all patch corners, we only have to enforce (11_1) for triangular and (9_1), (9_2) for rectangular patches to fulfill all requirements of the equality constraints, (4). We denote the middle Bézier coefficients of the boundary curve from P^i to P^j by b^{ij} and b^{ji}, with b^{ij} the Bézier coefficient closer to P^i.

For *triangular* patches this means solving the following 3×3 system for the center coefficient b:

$$\begin{pmatrix} n^1 \\ n^2 \\ n^3 \end{pmatrix} b = \begin{pmatrix} n^1P^1-\frac{3}{6}n^2(b^{13}-P^1) \\ n^2P^2-\frac{3}{6}n^3(b^{21}-P^2) \\ n^3P^3-\frac{3}{6}n^1(b^{32}-P^3) \end{pmatrix}. \tag{23}$$

The *rectangular* patches have 4 center coefficients available for the enforcement of the remaining 8 conditions. We can break the resulting 8×12 system into 4 pieces of 3 equations by choosing constants κ^i:

$$\begin{pmatrix} n^i \\ n^{i+1} \\ n^{i-1} \end{pmatrix} b = \begin{pmatrix} n^iP^i-\frac{3}{9}n^{i+1}(b^{ii-1}-P^i) \\ n^{i+1}b^{ii+1}+\kappa^i \\ n^{i-1}b^{ii-1}+\kappa^{i-1} \end{pmatrix} \quad \text{for} \quad i\in\{1,2,3,4\} \tag{24}$$

Theorem 3.4. *The system (24), for rectangular patches, is always solvable. If Assumption 3.2 holds, then (23) is uniquely solvable. If $n^i = n^j$, then (23) is solvable if and only if the data are planar along the edge ij, i.e. $\delta^{ij} n^i = \delta^{ij} n^j = 0$.*

Proof: First, we look at (23). If Assumption 3.2 holds then the left hand matrix has an inverse. If $n^i = n^j$ then the difference of the right hand sides of equations i and j must be zero, i.e.

$$n^i \delta^{ij} - \frac{1}{2}(n^i(b^{ik} - P^i) - n^k(b^{ji} - P^j) = 0, \tag{25}$$

where $k \neq i$ and $k \neq j$. By (10)

$$n^k(b^{ji} - P^j) = n^i(b^{jk} - P^j). \tag{26}$$

Since (S_0) implies that $n^l(b^{lk} - P^l) = 0$, (25) is equivalent to $\delta^{ij} n^i = \delta^{ij} n^j = 0$. The reverse argument shows that planar data yields a 1- or even 2-dimensional set of solutions.

With the choice $\kappa^i = -\frac{3}{9} n^{i+2}(b^{i+1i} - P^{i+1})$, (24) is solvable under the same conditions as (23). But, since we may choose the κ^i freely, (24) is in fact always solvable by adjusting the right hand side appropriately. ♠

4. Cusps and Curvature

We now analyze under what conditions on the data the inequalities (3) hold. That is, given that (12) is solvable, we ask under what conditions the scheme is also cusp-free. First, we look at the *data points.*

4.1. Cusps at the data points

At any data point P, (3) is equivalent to

$$\det(d^l d^{l-1}) > 0 \quad \text{for all} \quad l \in \{1, \ldots, k\}. \tag{27}$$

(Recall from Section 3.1 that d^i is the restriction of the i^{th} tangent vector at P to the tangent plane.) We will reformulate and express (27) in terms of the data. As in Section 3, the neighbors of a data point are numbered 1 through k and superscripts i not in the range $\{1, \ldots, k\}$ will be interpreted accordingly.

We continue the analysis from Theorem 3.1 to prove

Theorem 4.1. *If (12) is invertible, then the match between the patches abutting along boundary curve l is C^1 at P if and only if*

$$\frac{1}{D^{l-1}}((\frac{1}{2}D^{l-1}S^l_{odd})^2 - r^l r^{l-1}) > 0, \tag{28}$$

where the notation $S^l_{odd} := (-)^{l-1}(\sum_{j=1}^{l-1}(-)^j c^j r^j - \sum_{j=l}^{k}(-)^j c^j r^j)$ is consistent with (20) except for the modulo-induced sign change.

Proof: We continue the proof of Theorem 3.1 by backsolving. Using (12″) we can compute d^{i-1} in terms of d^i for $i \in \{2, \ldots, k\}$ as

$$d^{i-1} = \frac{1}{D^{i-1}} \begin{pmatrix} m_1^i & -m_1^{i-1} \\ -m_0^i & m_0^{i-1} \end{pmatrix} \begin{pmatrix} r^{i-1} \\ m^{i-1} d^i \end{pmatrix}. \tag{29}$$

This allows us to rewrite the left hand side expression in (27) as

$$\det(d^i d^{i-1}) = \frac{1}{D^{i-1}}((m^{i-1}d^i)^2 - (m^i d^i) r^{i-1}) = \frac{1}{D^{i-1}}((m^{i-1}d^i)^2 - r^i r^{i-1}). \tag{30}$$

We now need only d^k in terms of the data to be able to express all d^i and all conditions (27) in terms of the data. To compute d^k, we recall that $\gamma^1 \ldots \gamma^k = -1$ if (12) is invertible (Theorem 3.1), and thus $\widetilde{m} = m^{k-1}/\gamma^k + m^1$. We now solve (16) for d^k:

$$d^k = \frac{1}{2D^k} \begin{pmatrix} \widetilde{m}_1 & -m_1^k \\ -\widetilde{m}_0 & m_0^k \end{pmatrix} \begin{pmatrix} r^k \\ s \end{pmatrix} \tag{31}$$

and obtain

$$m^{k-1} d^k = \frac{1}{2D^k} (\, B^k, \; -D^{k-1}\,) \begin{pmatrix} r^k \\ s \end{pmatrix} = \frac{1}{2}(\beta^k r^k + \gamma^k s) = \frac{1}{2} D^{k-1}(c^k r^k + \sum_{1}^{k-1}(-)^j c^j r^j). \tag{32}$$

Thus (30) yields

$$\det(d^k d^{k-1}) = \frac{1}{D^{k-1}}((\frac{1}{2}D^{k-1}S^k_{odd})^2 - r^k r^{k-1}). \tag{33}$$

By (odd number of neighbors) symmetry, we arrive at the results for $l \neq k$. (Note that $S^l = S^{l+1} + 2(-)^l c^l r^l$.) ♠

Corollary 4.1. *If $D^{l-1} > 0$ and $r^l r^{l-1} < 0$ for all $l \in \{1, \ldots, k\}$ and k odd, then the match is cusp-free at the data point.*

$D^l > 0$ is not a very natural condition though, since it implies that the order of projected neighbor normals is counter-clockwise, i.e. opposite to the order of the neighbors.

We call **the neighborhood of P convex** if $D^i < 0$ for all $i \in \{1, \ldots, k\}$. For convex neighborhoods we have

Corollary 4.2. *If $c^i r^i = c^j r^j$ for all $i, j \in \{1, \ldots, k\}$ and k odd, and the neighborhood of P is convex, then (28) is equivalent to*

$$(\beta^l)^2 r^l < 4r^{l-1}.$$

We apply the Corollary in the important

Example: Consider points equally distributed on a sphere, such that each point has an odd number of neighbors. By symmetry $c^i = c^j$ and $r^i = r^j$ for the neighbors i and j of a point P. The surface is cusp-free since $r^i > 0$ and $\det(m^{i-1}, m^{i+1}) < \det(m^{i-1}, m^i) + \det(m^i, m^{i+1}) = 2\det(m^i, m^{i+1})$. ♠

If P has an even number of neighbors and (12) is solvable, we may impose an additional condition, for example on d^k. This idea leads to

Corollary 4.3. *Let Assumption 3.2 hold. If P has an even number of neighbors and the system (12) corresponding to P is solvable, then the match between the patches abutting along the boundary curve connecting P with P^l is C^1 at P if and only if*

$$\frac{1}{D^{l-1}}((D^{l-1}S^k_l)^2 - r^l r^{l-1}) > 0, \tag{34}$$

where $S^k_l := \sum_{j=l}^{k-1}(-)^j c^j r^j + \tau$ and τ is freely choosable.

Proof: Since (12′) has a 1-dimensional set of solutions we may set $m^k d^1 = \tau D^{k-1}$ in the last equation of (12′). Taking τ to the right hand side, we find, since Assumption 3.2 holds, that the system is uniquely solvable for τ. (29) yields the recurrence

$$d^i m^{i-1} = \beta^i r^i + \gamma^i(d^{i+1}m^i) = D^{i-1}(c^i r^i - \frac{1}{D^i}(m^i d^{i+1}).$$

We use the recurrence together with (30) to derive (34) by induction on i starting with $S^k_k = \tau$. ♠

Finally, given that (12) is solvable, each linearly dependent pair or triple of edge-adjacent normals allows us to add an additional constraint. If $D^i \neq 0$, we can add the constraint $d^i m^{i-1} = \tau^i$ in the spirit of Corollary 4.2.

More directly (cf. Section 3.1), with N the normal at P, $(N \times \delta^i)d^i = 0$ is a consistent choice for an additional equation, forcing d^i to lie on the projection of δ^i into the tangent plane at P. This condition is also appropriate for reproducing a flat surface.

4.2 Cusping not at the data points

Little beyond (3) can be said about cusping in the interior of the boundary curve. In our experience, cusping does not occur if (27) is satisfied with a good margin.

Note that the following approach to measuring cusping at the boundary is flawed. Consider three adjacent boundary curves that are consecutive in the neighbor list of the data point they share. Let the first be $s(v)$, the second $s(u)$ and the third $t(w)$ as in Section 2. Then (3) for triangular patches is in general *not equivalent* to

$$(n \times s_u)t_w > 0.$$

This is because the size of n plays a crucial role in the inequality and the size of n is not necessarily equal to the size of $s_v \times s_u$.

5. Curvature weights

Theorem 4.1 shows that cusping at a data point depends in general on the locations and normals of that data point and all its neighbors. We could change the normal at a data point. But this affects not only the k inequalities at the data point, but also all systems of equations asociated with the neighbors.

It turns out, though, that our choice of n in Section 3 was unnecessarily restrictive. Since we only have to interpolate to the normal *direction* at the data points, we may choose n as

$$n(u) = \{n^0, n^1\} := \{N^0, \omega N^1\},$$

where ω is a positive weight (that influences the curvature). Changes in the weight of a boundary curve affect only the systems of equations at either end.

We determine the weight ω^{ij} for the boundary curve from P^i to P^j in dependence on the data before computing any of the Bézier coefficients. That is, we prescribe the normal direction, just as in Section 3, and avoid global nonlinear equations. First we argue that $\omega^{ij} = \alpha^{ij}$ is a reasonable default value. Then we alter ω^{ij} to influence the cusping behavior.

We can compute the tangent vectors d^i in (12) as perturbations of the [bi]linear interpolant. The [bi]linear interpolant (equivalent to a cubic with $d^i = 0$ for all i) does not cusp (but is, of course, in general not C^1). Thus we want the perturbation from $d^i = 0$ to be small, i.e. the right hand side terms r^i of (12) to be as close to 0 as possible. Setting $r^j = 0$ at P^i, we get $\omega^{ij} = \frac{1}{2}\alpha^{ij}$. Setting $r^i = 0$ at P^j, we get $\omega^{ij} = 2\alpha^{ij}$. Thus $\omega^{ij} = \alpha^{ij}$ seems a natural compromise.

For a second argument, we consider fitting a *quadratic* boundary curve $\{P^i, 2b, P^j\}$ with derivative $\{d^0, d^1\}$ to the data. It turns out that $\omega^{ij} = \alpha^{ij}$ is a necessary condition for

$$ns_u = \{n_0 d_0, n_0 d_1 + n_1 d_0, n_1 d_1\} = 0, \tag{35}$$

since (35_0), (35_1) and (35_2) imply

$$n^0 P^1 - n^1 P^0 = n^0 b - n^1 b = n^0 P^0 - n^1 P^1$$

or, equivalently,

$$N^0 \delta^{10} = \omega^{01} N^1 \delta^{01}.$$

We note that the choice $\omega^{ij} = \alpha^{ij}$ simplifies r^i: $r^i = -\frac{2}{3}\delta^i N$.

Next we show how an increase of ω^l by some positive factor Δ^l affects the cusping conditions. We mark the perturbed quantities with a Δ. Then $c^i_\Delta = c^i$ and $r^i_\Delta = r^i$ for $i \neq l$, while c^l and r^l become

$$c^l_\Delta = (\frac{1}{\Delta^l})^2 \frac{B^l}{D^{l-1} D^l} \quad \text{and} \quad r^l_\Delta = \frac{2}{3}\delta^l (N + 2\Delta^l N^l).$$

Since D^{l-1} and D^l increase by Δ^l, the new inequalities in place of (28) read:

$$(28_a) \qquad \frac{1}{D^{i-1}}((\frac{1}{2}D^{i-1}\Delta^i S^i_\Delta)^2 - r^i_\Delta r^{i-1}_\Delta) > 0 \quad \text{for } i \in \{l, l+1\},$$

$$(28_b) \qquad \frac{1}{D^{i-1}}((\frac{1}{2}D^{i-1}S^i_\Delta)^2 - r^i r^{i-1}) > 0 \quad \text{for } i \notin \{l-1, l\},$$

where

$$S^i_\Delta := (-)^i(\sum_{j=i+1}^{k} (-)^j c^j r^j + (-)^i c^i_\Delta r^i_\Delta - \sum_{j=1}^{i-1} (-)^j c^j r^j).$$

The effect is thus concentrated on the four inequalities associated with the boundary l.

6. Examples

Having laboured through three sections of algebra it is time to reap the harvest. The first 4 examples displayed below were generated with $n := \{N^0, N^1\}$. All other examples use $\omega = \alpha$ unless mentioned otherwise. Occasionally, parts of the coordinate axes are visible in the rendered images.

The first sequence of examples aims at approximating a sphere. 4,6,8 and 12 points, evenly distributed on a sphere, form the input sequence for Figures 6.a to 6.d. We will refer to the data by the shape of their piecewise linear interpolant.

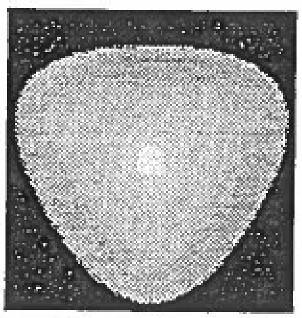

Figure 6.a: Tetrahedron (4 points).

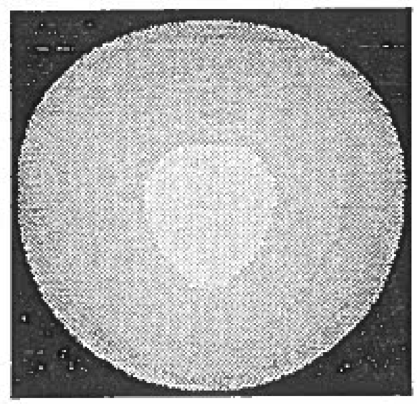

Figure 6.b: Cube (6 points).

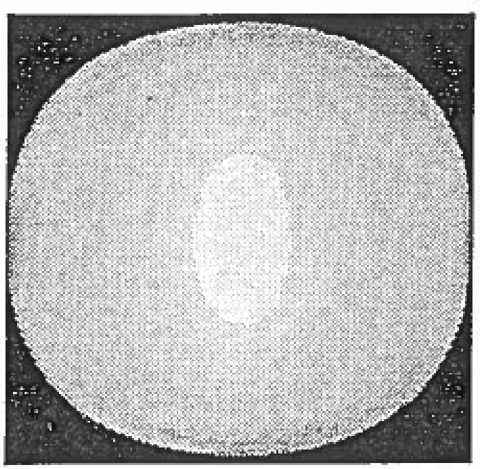

Figure 6.c: Octahedron (8 points).

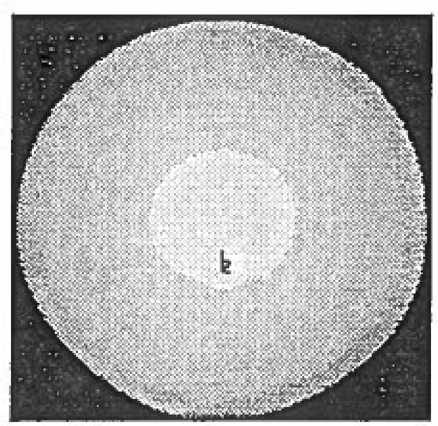

Figure 6.d: Dodecahedron (12 points).

The ratio of the volumes enclosed by the approximate spheres to that of a perfect sphere with the same radius are 1.558, 1.102, 1.161 and 1.043 respectively. The maximal curvatures are (approximately) 1.323, 1.03, 1.152 and 1.059. The cube consists entirely of rectangular patches, whereas the other data objects have a triangulated surface.

Next, we show two surfaces that combine rectangular and triangular patches.

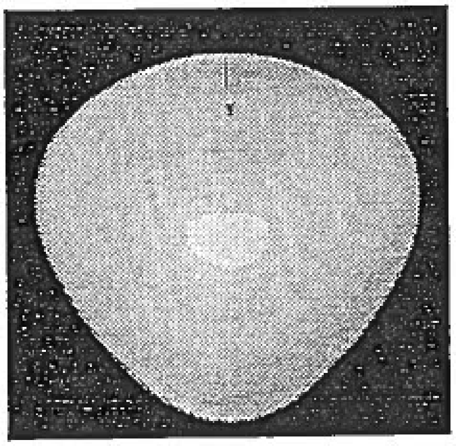

Figure 6.e: Pyramid (5 points).

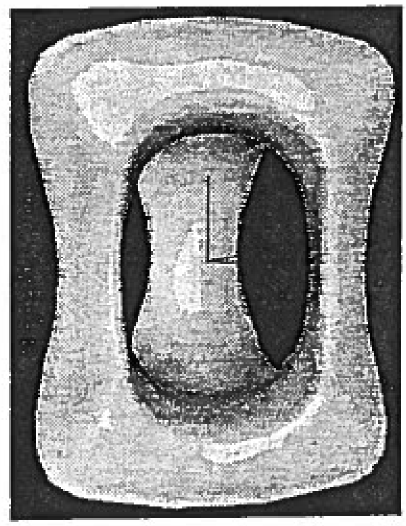

Figure 6.f: 3 semi-doughnuts meeting twice at 120°. (50 points).

Finally, we present an open surface. The saddle consists of 4 triangular patches.

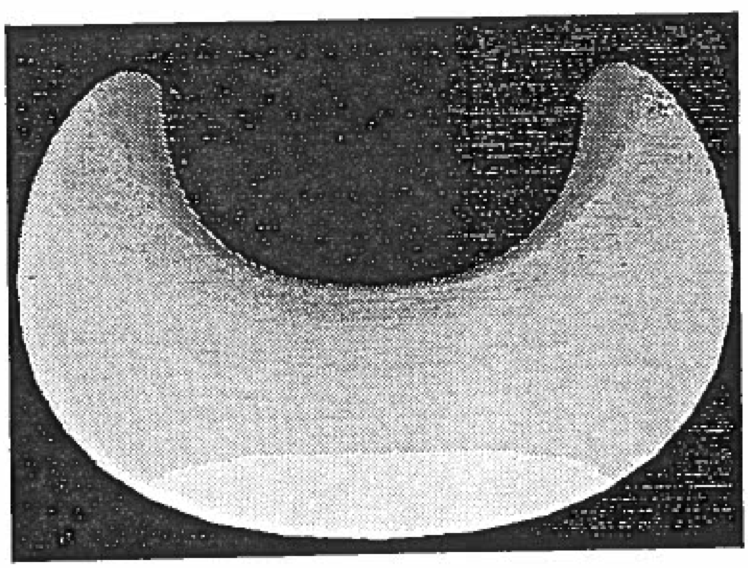

Figure 6.g: Saddle (5 points).

We demonstrate the effect of the curvature weights on the shape (and prevention of cusps) in Figures 6.h and 6.i. The top of the regular octahedron (Figure 6.c) is pushed in. The left hand side shows the cusping match without curvature weights, the right hand side the C^1 match with $\omega = \alpha$. The surface remains C^1 until the data are no longer convex, i.e. until the top is pushed below the plane spanned by its four neighbors.

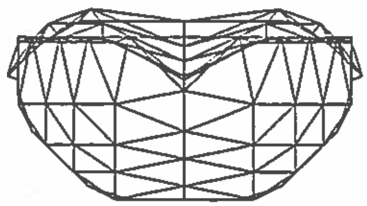

Figure 6.h: "Squashed" octahedron without curvature weights.

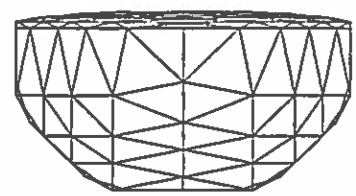

Figure 6.i: ... with curvature weights.

To illustrate Section 4 and show the effect of perturbations of the normal, the next sequence shows the evolution of a cusp. The Bezier net (twice refined) corresponds to one of the 8 octahedron patches. We slant the normal at the top of the patch in the direction of the two other data points until the order of the tangent vectors at those data points is reversed (Figure 6.l). The normal directions are (.3,0,1), (.4,0,1) and (.5,0,1), respectively, with the data points at (0,0,1) (top), (.707,.707,0) and (.707,-.707,0).

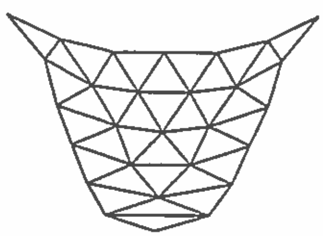

Figure 6.j: early stage.

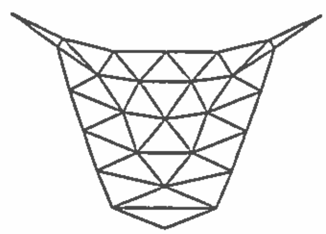

Figure 6.k: almost a cusp.

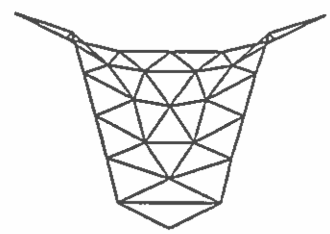

Figure 6.l: 2 cusps.

The damage can be repaired by adjusting the weights along the boundaries. The result is the skewed octahedron in Figure 6.m below.

We conclude with a detail of the BB-net of Figure 6.f.

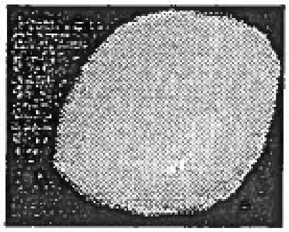

Figure 6.m: Skewed octahedron.

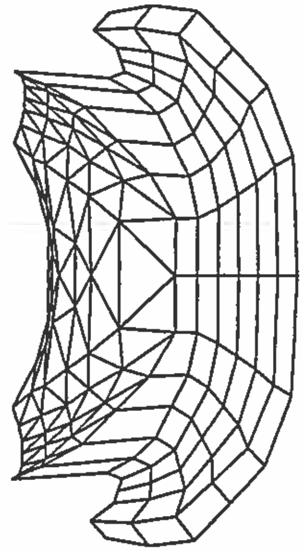

Figure 6.i: A BB-net detail of Figure 6.f. (5 patches)

7. Conclusions

The paper leaves several questions unresolved.

- Is there a grid refinement or a strategy to choose normals such that Assumption 3.1 holds , i.e. such that the shapes of the neighborhoods at adjacent points P^i and P^j always agree?
- What would be a more intuitive (sufficient) characterization of the cusping conditions of Theorem 4.1? In particular, what is the relationship between the data-implied curvature and the quantities $c^l r^l$?

Other approaches have been looked at.

- A cubic scheme with a quadratic normal leads to nonlinear systems of equations. Sabin's cubic scheme for rectangular patches [Sabin68], for example, introduces small local nonlinear systems of equations by prescribing the curvature at the data points. The conditions for which these systems are solvable have not been clearly characterized in terms of the data.
- An analogous quartic scheme with a data-dependent quadratic boundary normal leads to a global system of equations for the middle term of n.

Acknowledgements: I thank Carl de Boor for his many helpful comments.

References

de Boor, Carl (1987), B-form basics, *Geometric Modeling*, G. Farin ed., SIAM.

Farin, Gerald (1986), Triangular Bernstein-Bézier patches, *Computer Aided Geometric Design* 3.

Sabin, M.A. (1968), Conditions for continuity of surface normal between adjacent parametric surfaces, *Tech. Rep., British Aircraft Corporation Ltd.* .