Chapter 8

A Survey of Parametric Scattered Data Fitting Using Triangular Interpolants

This paper has been published as a chapter in "Curve and Surface Design", H. Hagen, (ed), SIAM, 1992
Some of the figures from that paper are missing from this version, as are all of the black-and-white photographs.

There are currently a number of methods for solving variants of the following problem: Given a triangulated polyhedron P in three space with or without boundary, construct a smooth surface that interpolates the vertices of P. In general, while the methods satisfy the continuity and interpolation requirements of the problem, they often fail to produce pleasing shapes. The purpose of this paper is to present a unifying survey of the published methods, to identify causes of shape defects, and to offer suggestions for improving the aesthetic quality of the interpolants.

8.1 Introduction.

The problem of passing a surface through a set of data points arises in numerous areas of application such as medical imaging, geological modeling, scientific visualization, and geometric modeling. Variants of this problem have been approached from many directions. Tensor-product B-splines work well for modeling surfaces based on rectilinear control nets but are not sufficient for more general topologies. Triangulated data, however, can represent

arbitrary topologies. In this paper, we present a survey of a class of schemes that address the problem of fitting a surface to triangulated data.

One way to categorize surface fitting schemes is by the locality of data used in constructing a portion of the surface. A global scheme will use arbitrarily many of the data points in constructing each portion of the surface; a local scheme will only consider those points near the portion of the surface it is creating. Only local schemes are considered in this paper.

If the surface to be constructed lies above the plane it can be described as $\mathbf{S}(x, y) = (x, y, f(x, y))$. The data set is then referred to as scalar data, as the surface can be thought of as a scalar valued function over the plane. Such data can be interpolated with a C^1 surface, using, for instance, the methods surveyed by Barnhill [1] and Franke [11].

A parametric scheme, on the other hand, constructs a vector valued surface, $\mathbf{S}(u, v) = (x(u, v), y(u, v), z(u, v))$ and, unlike a scalar method, is capable of representing arbitrary topologies. The parametric problem is generally considered to be more difficult than the scalar variant. It has been shown, for instance, that the data cannot always be interpolated with a parametrically continuous surface [19]. Instead, the continuity conditions have to be relaxed to G^1 (tangent plane) continuity. The schemes surveyed in this paper are all parametric schemes.

In addition to geometric data, parametric interpolation schemes require information about the topology of the desired surface. The topological information is usually specified as adjacency information relating the data points (vertices), edges, and faces. The schemes we have considered all assume that faces are triangular (i.e., three bounding edges per face), and that any number of faces may join at a vertex. These "triangular meshes" are sufficiently general to represent arbitrary topological surfaces.

Finally, surface fitting schemes may interpolate or approximate the given data. Interpolating schemes construct surfaces that pass through the given data points. Approximating schemes produce surfaces that retain the topology of the input data, but only pass near the data points. For some applications, an interpolating scheme is preferred, while for other applications, an approximating scheme may be a better choice. Here we will only consider interpolating schemes, and we will ignore the issue of specialized boundary conditions.

Thus, the primary goal of this paper is to present a unifying survey of local, parametric, triangular, interpolatory data fitting schemes. The surveyed schemes all proceed by first building boundary curves for a face and then filling in the interior of the face with one or more surface patches.

While all the schemes surveyed meet mathematical smoothness conditions, none of them produces surfaces with pleasing shape. Further, despite the diversity of the methods, all the schemes we implemented produced similar shape defects. Our investigations indicate that these poor shapes are primarily an artifact of the construction of boundary curves.

In Section 8.3, we present some background material. Three methods of constructing a tangent plane continuous join between two patches are presented in Section 8.4. In Section 8.5, the surveyed schemes are described. In Section 8.6, we look at the surfaces produced by these schemes and consider ways of improving their shapes. In Section 8.7, we summarize and present some recommendations.

8.2 Notation.

Throughout this paper, scalars and scalar valued functions will be denoted by *non-bold type letters* and Greek letters, such as r and α. Points and point valued functions will be denoted with boldface letters, such as $\mathbf{V}$. Vectors will be represented by boldface letters topped with an arrow, such as $\vec{\mathbf{T}}$. Unit vectors will be denoted as $\hat{\mathbf{C}}$.

Surface patches will be denoted by the boldface letters $\mathbf{F}$ and $\mathbf{G}$. Usually, these surface patches will be in triangular Bézier form [10]. Often, we will consider the case when $\mathbf{F}$ and $\mathbf{G}$ are adjacent patches. In this case, the control points associated only with patch $\mathbf{F}$ will be denoted by $\mathbf{F}_i$, the control points associated only with patch $\mathbf{G}$ will be denoted by $\mathbf{G}_i$, and the control points common to both patches will be denoted by $\mathbf{H}_i$.

The vertices of a triangle in the domain of a patch will be denoted by $\mathbf{p}$, $\mathbf{q}$, and $\mathbf{r}$. The corresponding vertices in the range will be described by $\mathbf{V}_P$, $\mathbf{V}_Q$, and $\mathbf{V}_R$.

The directional derivative of a surface $\mathbf{F}$ in the direction $\vec{\mathbf{r}}$ is denoted by $\mathbf{D}_{\vec{\mathbf{r}}}\mathbf{F}$. The derivative of a parametric curve $\mathbf{H}(t)$ is denoted $\mathbf{H}'(t)$. We will have use for a certain radial direction in the triangle $\mathbf{pqr}$, namely, $\vec{\mathbf{r}}_{\mathbf{p}}(t) = ((1-t)\mathbf{q} + t\mathbf{r}) - \mathbf{p}$.

Finally, $B_i^n(t)$ will denote the ith Bernstein polynomial of nth degree, i.e.,

$$B_i^n(t) = \left(\begin{array}{c} n \\ i \end{array} \right) (1-t)^{n-i} t^i.$$

8.3 Background.

Local interpolation schemes generally construct a surface consisting of multiple surface patches. In order for the entire surface to look smooth, certain continuity conditions must be met at every boundary between two patches. To avoid holes in the surface, every pair of neighboring patches must meet with C^0 continuity. To ensure that adjacent patches meet smoothly, one might also want them to meet with a continuous first derivative. However, this is not possible for surfaces of arbitrary topology. An alternate approach is to construct the surface patches to meet with continuous tangent planes along the boundaries. The patches are then said to meet with G^1 continuity (cf. [2, 20]). Several methods of ensuring tangent plane continuity will be presented in Section 8.4.

A second issue is what is sometimes referred to as the *vertex consistency problem.* This problem occurs when trying to construct a single C^2 patch for each triangular face of the data. The G^1 continuity conditions between patches set up a system of constraints around each data point. For a vertex of even degree greater than four, it has been shown that this system can not necessarily be satisfied [32].

There are primarily two approaches taken to avoid this problem. The first approach constructs multiple patches per face, which successfully decouples the cycle of constraints. A second approach is to construct patches that are not C^2 at the data points. The schemes surveyed in this paper all use one of these two approaches.

A third approach to solving the vertex consistency problem is to construct a "C^2 consistent" curve network. Peters has shown that if the boundary curves adjacent to a data point all agree with a common second fundamental form, then the above mentioned cycle of constraints can be satisfied [28]. Note that while this is a sufficient condition for satisfying the vertex consistency problem, it is not a necessary condition. An alternate approach to constructing a C^2 consistent curve network can be found in [23].

8.4 Tangent Plane Continuity.

Fitting surface patches together with tangent plane continuity has been approached from several directions. Farin [7] and Piper [29] give sufficient conditions for two polynomial patches to meet with G^1 continuity. A second approach, taken in [4, 19, 21, 26], is to first create a cross boundary tangent

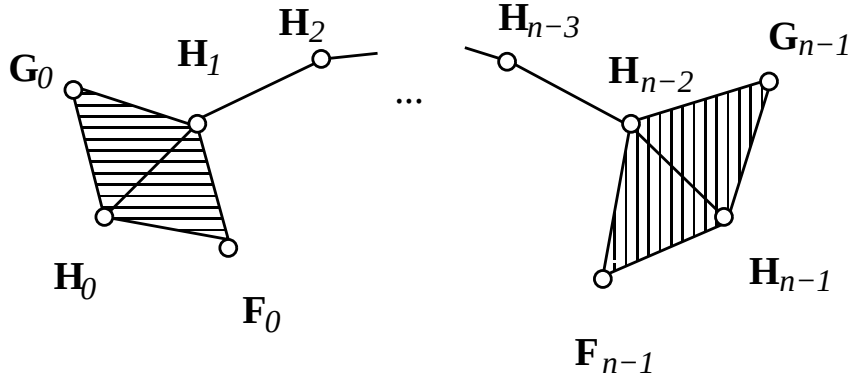

Figure 8.1: Bézier control points used in Farin's G^1 conditions.

vector field for each boundary and then to construct patches that agree with these cross boundary fields.

8.4.1 Farin.

Farin [7] presented conditions for two degree n polynomial patches with a common degree $n-1$ boundary to meet with G^1 continuity. Labeling the Bézier control points as in Figure 8.1, Farin's conditions are as follows.

Given two degree n polynomial patches with a common degree $n-1$ boundary, where

$$\begin{array}{rcll} \mathbf{G}_0 & = & \alpha_1\mathbf{H}_0 + \alpha_2\mathbf{H}_1 + \alpha\mathbf{F}_0, & \alpha_1+\alpha_2+\alpha = 1, \\ \mathbf{G}_n & = & \alpha_3\mathbf{H}_{n-1} + \alpha_4\mathbf{H}_n + \alpha\mathbf{F}_n, & \alpha_3+\alpha_4+\alpha = 1, \end{array} \tag{8.1}$$

then the two patches meet with G^1 continuity if

$$\mathbf{G}_i = \frac{n-i}{n}(\alpha_1\mathbf{H}_i + \alpha_2\mathbf{H}_{i+1} + \alpha\mathbf{F}_i) + \frac{i}{n}(\alpha_3\mathbf{H}_{i-1} + \alpha_4\mathbf{H}_i + \alpha\mathbf{F}_i).$$

Equation 8.1 can be formulated in terms of the areas of the shaded triangles of Figure 8.1:

$$\frac{\text{area}(\mathbf{G}_0,\mathbf{H}_0,\mathbf{H}_1)}{\text{area}(\mathbf{F}_0,\mathbf{H}_0,\mathbf{H}_1)} = \frac{\text{area}(\mathbf{G}_{n-1},\mathbf{H}_{n-1},\mathbf{H}_{n-2})}{\text{area}(\mathbf{F}_{n-1},\mathbf{H}_{n-1},\mathbf{H}_{n-2})}.$$

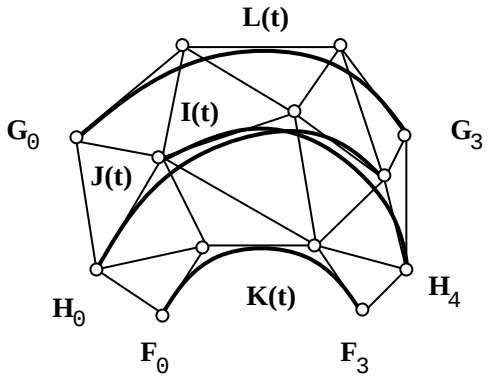

Figure 8.2: The curves used in Piper's G^1 conditions.

8.4.2 Piper.

Piper [29] develops sufficient conditions for two quartic patches with quartic boundaries to meet with G^1 continuity. He begins by noting that the following equation must hold for patches $\mathbf{F}$ and $\mathbf{G}$ to meet G^1:

$$e(t)\mathbf{I}(t) + f(t)\mathbf{J}(t) + g(t)\mathbf{K}(t) + h(t)\mathbf{L}(t) = \vec{\mathbf{0}}, \tag{8.2}$$

where e, f, g, and h are scalar functions such that

$$e(t) + f(t) + g(t) + h(t) = 0$$

for $t \in [0, 1]$, and where

$$\mathbf{I}(t) = \sum B_i^3(t)\mathbf{H}_{i+1},$$

$$\mathbf{J}(t) = \sum B_i^3(t)\mathbf{H}_i,$$

$$\mathbf{K}(t) = \sum B_i^3(t)\mathbf{F}_i,$$

$$\mathbf{L}(t) = \sum B_i^3(t)\mathbf{G}_i.$$

These curves are illustrated in Figure 8.2. Here, the points $\mathbf{F}_0$, $\mathbf{F}_3$, $\mathbf{G}_0$, $\mathbf{G}_3$, $\mathbf{H}_0$, $\mathbf{H}_1$, $\mathbf{H}_3$, and $\mathbf{H}_4$ are known.

Piper restricts e, f, g, and h to be linear functions. Equation 8.2 then reduces to a 3×5 system of linear equations, where the unknowns are the control points $\mathbf{F}_1$, $\mathbf{F}_2$, $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{H}_2$. In certain situations, the functions e, f, g, and h are all constant functions. In this case, the 3×5 system of

equations reduces to a 2×5 system. In summary, these constraints represent underdetermined conditions on the unknown control points to achieve a G^1 join of the patches.

8.4.3 Chiyokura-Kimura, Herron, Jensen.

As mentioned earlier, one approach to creating surface patches that meet with G^1 continuity is to first construct a field of cross boundary tangent vectors along the boundary between two patches, using data common to both patches. The cross boundary tangent field, together with the first derivative vector of the boundary curve, defines a tangent plane field all along the boundary. Next, the two patches are constructed, one on either side of the boundary, that match this tangent plane field along the boundary. The two patches will therefore meet with G^1 continuity. This is the approach taken in [4, 19, 21, 26]. We present now the method of Chiyokura and Kimura and show its relationship to other constructions.

Although Chiyokura and Kimura's cross boundary construction was originally intended for rectangular patches, it readily extends to the construction of quartic triangular Bézier patches [31]. The method uses the boundary data for two adjacent patches to construct the Bézier control points that influence the tangent plane behavior along their common boundary. The boundary data consist of a cubic polynomial boundary curve and a pair of tangent vectors at each end of the boundary curve (Figure 8.3 shows this data in Bézier form). Two quartic interior Bézier control points for each patch are set so as to match the tangent plane field. We will give the construction for only one patch since the construction for the other patch is identical.

The cross boundary tangent vector field is defined by linearly blending two vectors, one in each of the tangent planes at the end points. Chiyokura and Kimura choose these vectors to be unit vectors perpendicular to the tangents at the end points of the boundary curve. For patch $\mathbf{F}$, this blend is given by

$$\vec{\mathbf{C}}(t) = (1-t)\hat{\mathbf{C}}_0 + t\hat{\mathbf{C}}_1.$$

The two perpendiculars $\hat{\mathbf{C}}_0$ and $\hat{\mathbf{C}}_1$ are unique, up to sign. The signs are chosen based on the vectors $\vec{\mathbf{F}}_0$ and $\vec{\mathbf{F}}_3$ as shown below, where $\vec{\mathbf{F}}_i = \mathbf{F}_i - \mathbf{H}_i$. $\vec{\mathbf{C}}(t)$ together with $\mathbf{H}'(t)$ completely specifies the tangent plane field along the boundary.

For $\mathbf{F}$ to agree with the tangent plane field given by $\mathbf{H}'(t)$ and $\vec{\mathbf{C}}(t)$,

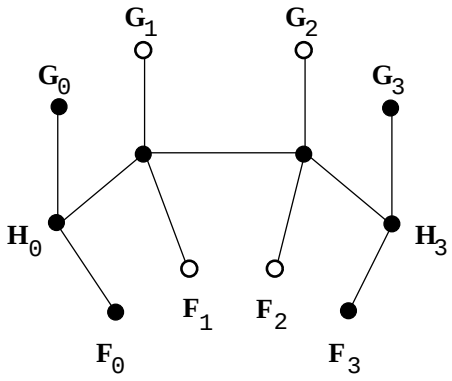

Figure 8.3: Boundary data used by Chiyokura and Kimura's construction. The solid points are the known control points; the hollow points are constructed so that the two patches meet G^1. Note that the **H**s are cubic control points, while the **F**s and **G**s are quartic control points.

there must exist functions $k(t)$ and $h(t)$ such that

$$\mathbf{D}_{\vec{\mathbf{r}}(t)}\mathbf{F}(0,t,1-t) = k(t)\cdot\vec{\mathbf{C}}(t) + h(t)\cdot\mathbf{H}'(t), \tag{8.3}$$

where $\vec{\mathbf{r}}(t)$ is the radial direction in the domain of **F**.

The values of $k(t)$ and $h(t)$ can be determined at the end points by evaluating Equation 8.3 at $t = 0$ and $t = 1$:

$$\vec{\mathbf{F}}_0 = k_0\cdot\hat{\mathbf{C}}_0 + h_0\cdot\vec{\mathbf{H}}_0,$$

$$\vec{\mathbf{F}}_3 = k_1\cdot\hat{\mathbf{C}}_1 + h_1\cdot\vec{\mathbf{H}}_2,$$

where $\vec{\mathbf{H}}_i = \mathbf{H}_{i+1} - \mathbf{H}_i$, $k_0 = k(0)$, $k_1 = k(1)$, $h_0 = h(0)$, and $h_1 = h(1)$. For h and k to interpolate these end point conditions, they both must be at least linear functions. If we restrict them to be no more than linear, then each is uniquely determined:

$$k(t) = k_0\cdot(1-t) + k_1\cdot t,$$

$$h(t) = h_0\cdot(1-t) + h_1\cdot t.$$

Rewriting Equation 8.3 in the cubic Bernstein basis, we can use the coefficients to $B_1^3(t)$ and $B_2^3(t)$ to determine the desired interior control points, resulting in:

$$\mathbf{F}_1 = \frac{1}{3}\{(k_0+k_1)\hat{\mathbf{C}}_0 + k_0\hat{\mathbf{C}}_1 + 2h_0\vec{\mathbf{H}}_1 + h_1\vec{\mathbf{H}}_0\} + \mathbf{H}_1, \tag{8.4}$$

$$\mathbf{F}_2 = \frac{1}{3}\{k_1\hat{\mathbf{C}}_0 + (k_0 + k_1))\hat{\mathbf{C}}_1 + h_0\vec{\mathbf{H}}_2 + 2h_1\vec{\mathbf{H}}_1\} + \mathbf{H}_2. \tag{8.5}$$

There is still some freedom left in Equation 8.3. If $h(t)$ is a linear function, then the product $k(t) \cdot \vec{\mathbf{C}}(t)$ must be a polynomial of no higher than cubic degree. In the above formulation, this product is only a quadratic polynomial. Either $k(t)$ or $\vec{\mathbf{C}}(t)$ could be increased from a linear function to a quadratic function. Increasing $k(t)$ to a quadratic function gives a scalar degree of freedom, while increasing the degree of $\vec{\mathbf{C}}(t)$ yields a vector degree of freedom. Jensen [21] used the former generalization. He used the same linear blend of unit vectors for $\vec{\mathbf{C}}(t)$, but used the following quadratic scale function:

$$k^*(t) = k_0 \cdot u_0(t) + C \cdot \frac{(k_0 + k_1)}{2} u_1(t) + k_1 \cdot u_2(t),$$

where

$$u_0(t) = 2t^2 - 3t + 1, \quad u_1(t) = 4t - 4t^2, \quad u_2(t) = 2t^2 - t,$$

and C is a scalar shape parameter.[1] For $C = 1$, $k^*(t) = k(t)$.

A second degree of freedom in Equation 8.3 is in the choice of $\hat{\mathbf{C}}_0$ and $\hat{\mathbf{C}}_1$. These two vectors may be chosen in any fashion that uses information available to both patches, where the construction from both sides gives the same vectors with opposite sign. For example, in a later paper [3], Chiyokura defines $\hat{\mathbf{C}}_0$ and $\hat{\mathbf{C}}_1$ as

$$\hat{\mathbf{C}}_0 = \frac{\mathbf{G}_0 - \mathbf{F}_0}{|\mathbf{G}_0 - \mathbf{F}_0|},$$

$$\hat{\mathbf{C}}_1 = \frac{\mathbf{G}_3 - \mathbf{F}_3}{|\mathbf{G}_3 - \mathbf{F}_3|}.$$

Note that this definition of $\hat{\mathbf{C}}_0$ and $\hat{\mathbf{C}}_1$ is affine invariant and requires knowledge about both patches neighboring the boundary, whereas the earlier definition is not affine invariant and only uses information about the boundary curve.

Although Herron [19] approaches the problem somewhat differently, his construction and the Chiyokura-Kimura construction build the same field of cross boundary tangent vectors along the boundary curves.

[1] In Jensen's paper, u_1 is given as $u_1(t) = 4t - t^2$. However, without the factor of 4 scaling t^2, $k^*(1)$ does not interpolate k_1.

8.5 Parametric Schemes.

The schemes studied in this survey fall into two categories: the split domain schemes and the convex combination schemes. The split domain schemes surveyed were presented in [21, 29, 31]. The convex combination schemes surveyed were presented in [19, 22, 26]. All methods surveyed build a surface for each triangular face by first computing boundary curves around the triangle, and then constructing one or more patches that match this boundary data. The two categories differ in their solution to the vertex consistency problem.

8.5.1 Split Domain Schemes.

An extensive body of literature exists that discusses the properties of polynomial Bézier patches (cf. Farin [10]). If the data could be fit with Bézier patches, we could draw on this body of knowledge to compute various properties of the surface. However, if the boundary curves are constructed independently of each other, then a single Bézier patch cannot in general be used to interpolate the data, as the vertex consistency problem cannot in general be solved. Split domain schemes avoid this problem by constructing three patches per face, essentially splitting the domain triangle into three subtriangles, as was done by Clough and Tocher for scalar valued data [9].

After splitting, each of the subpatches is used to interpolate the data along one of the boundaries. Splitting allows the data along each boundary to be matched independently of the data on the other two boundaries. The remaining degrees of freedom are used to make the internal boundaries of the three subpatches meet with G^1 continuity.

All three of the split domain schemes presented here construct quartic polynomial patches. Figure 8.4 schematically shows a labeling of the Bézier control points of these patches. Some of the schemes in this section compute cubic boundaries, so we will need to refer to both the cubic control points and the quartic control points for these boundaries. Symbols such as $\mathbf{I}^4$ and $\mathbf{E}^4$ refer to the quartic control points, whereas $\mathbf{I}^3$ and $\mathbf{E}^3$ refer to the corresponding cubic control points. The appearance of the resulting formulas is unfortunately visually complex; we have chosen it so that we can be specific about which quantities are to be used in the calculations.

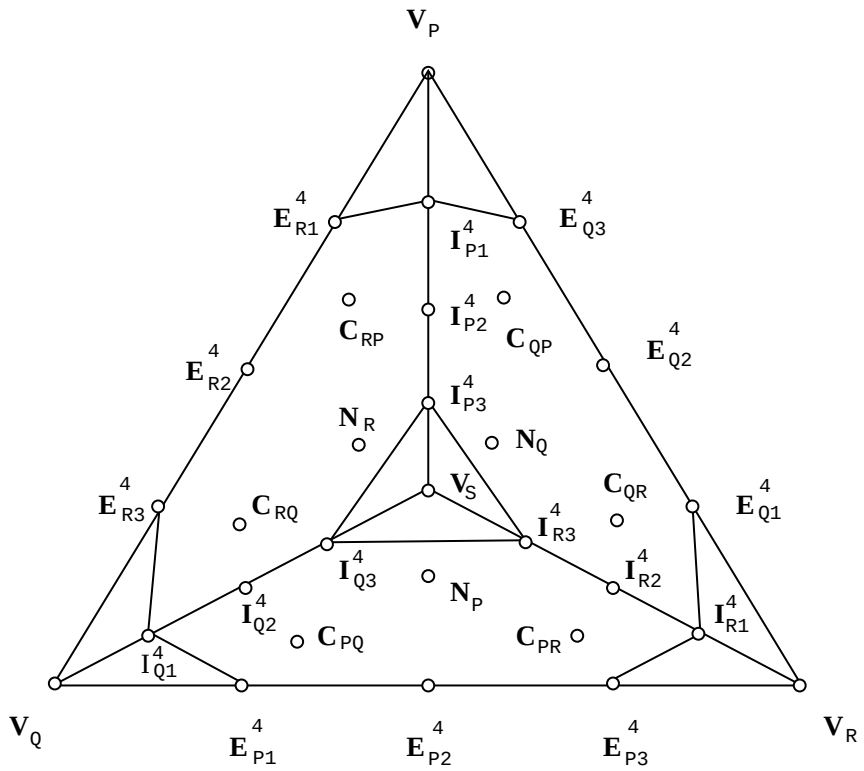

Figure 8.4: Bézier control points for split domain schemes.

Shirman-Sequin.

The following split domain scheme was proposed by Shirman and Sequin [31]. Three quartic triangular patches are constructed per face so as to interpolate the data points. The construction assumes that cubic boundary curves have been constructed and subsequently degree raised [10] to quartics.

Farin's G^1 conditions are used on the internal boundaries to ensure that the three patches meet each other with G^1 continuity. For $ijk \in \{PQR, QRP, RPQ\}$, the following relationships between the points $\mathbf{I}^3_{i1}$, $\mathbf{V}_i$, $\mathbf{E}^3_{i1}$, and $\mathbf{E}^3_{i2}$ are thus imposed:

$$\mathbf{E}^4_{j3} = \alpha_{i2}\mathbf{I}^3_{i1} + \alpha_{i1}\mathbf{V}_i + \alpha_i\mathbf{E}^4_{k1}, \tag{8.6}$$

where $\alpha_{i1} + \alpha_{i2} + \alpha_i = 1$. Similarly, at the other end of the internal boundaries, the following relationships hold:

$$\mathbf{I}^4_{k4} = \alpha_{i3}\mathbf{I}^3_{i3} + \alpha_{i4}\mathbf{S} + \alpha_i\mathbf{I}^4_{j4}, \tag{8.7}$$

where $\alpha_{i3} + \alpha_{i4} + \alpha_i = 1$. For reasons of symmetry, we set $\alpha_i = -1$ for all i.

A setting of the αs determines the control points $\mathbf{I}^3_{i1}$ according to Equation 8.6. The method of Chiyokura and Kimura can now be used to establish tangent plane continuity across the external boundaries (Equations 8.4 and 8.5), thus determining six of the interior control points (the $\mathbf{C}_{ij}$s).

Using these αs, Farin's continuity conditions set up a system of equations involving the interior control points which has the following solution:[2]

$$\mathbf{I}^3_{i2} = -\frac{\alpha_{i3}}{2\alpha_{i2}}\mathbf{V}_i - \left(\frac{\alpha_{i1}}{\alpha_{i2}} + \frac{\alpha_{i4}}{2\alpha_{i2}}\right)\mathbf{I}^3_{i1} + \frac{3}{2\alpha_{i2}}\left(\mathbf{C}_{ji} + \mathbf{C}_{ki}\right),$$

$$\begin{aligned}\mathbf{N}_i \;=\; & -\frac{\alpha_{i3}}{3}\mathbf{I}^3_{i1} + \frac{\alpha_{i3}}{3}(\mathbf{I}^3_{j1} + \mathbf{I}^3_{k1}) + \left(\frac{\alpha_{i2}}{18} - \frac{\alpha_{i1} + 2\alpha_{i4}}{6}\right)\mathbf{I}^3_{i2} + \\ & \left(\frac{\alpha_{i2}}{18} + \frac{\alpha_{i1} + 2\alpha_{i4}}{6}\right)\left(\mathbf{I}^3_{k2} + \mathbf{I}^3_{j2}\right),\end{aligned}$$

for $ijk \in \{PQR, QRP, RPQ\}$. Setting $\mathbf{S}$ to be the centroid of the $\mathbf{I}^3_{i2}$s fixes the following αs:

$$\alpha_{i3} = -\frac{3}{4},\ \alpha_{i4} = \frac{11}{4}.$$

[2] An error was made in the published version of these equations. Here we present a correct solution.

α_{i1} and α_{i2} are now related by $\alpha_{i1} + \alpha_{i2} = 2$. This leaves a scalar shape parameter to influence the shape of the interior of the patches. By setting $\alpha_{i1} = -\frac{1}{4}$ and $\alpha_{i2} = \frac{9}{4}$ the $\mathbf{I}^3_{i1}$s will be placed as in Shirman and Sequin's paper (i.e., $\mathbf{I}^3_{i1}$ will lie at the centroid of the triangle $\mathbf{V}_i\mathbf{E}^3_{j1}\mathbf{E}^3_{k3}$).

Jensen.

Except for two differences, Jensen's method [21] is quite similar to the method of Shirman and Sequin. The first difference is in the construction of the cross boundary tangent vector field along the boundaries. As mentioned earlier, both methods compute a linearly varying cross boundary tangent vector field. However, Jensen then uses a quadratic scaling function instead of the linear one used by Shirman-Sequin and others. The second way in which Jensen's method differs from Shirman and Sequin's method is in the construction of the interior boundaries. While Shirman and Sequin construct their patches to meet with G^1 continuity, Jensen uses C^1 conditions in the construction of the interior points.

Piper.

Piper's construction differs somewhat from that of the above two split domain schemes. First, a single cubic patch is constructed for each face. Next, this cubic patch is subdivided at the centroid into three cubic subpatches. These patches are then modified to produce "candidate" patches, which are degree elevated to quartic patches. The control points of the quartic patches are adjusted so that they satisfy the tangent plane continuity conditions of Section 8.4.2 along the exterior boundaries. As there are more than one set of such control points that satisfy Piper's continuity conditions (due to rank deficiency of the linear system), the set chosen is the one closest to the candidate control points in a least squares sense. Finally, the control points along the interior boundaries are adjusted so that the three patches meet each other with C^1 continuity.

8.5.2 Convex Combination Schemes.

Convex combination schemes create a single patch for each face. The patches are C^2 everywhere except at the vertices. This successfully avoids the vertex consistency problem by not having consistently defined mixed partial (i.e., *twist*) terms at the patch corners. Each patch is constructed by first building boundary curves and tangent plane fields along these boundary

curves. Next, three patches are created, each of which interpolates part of the boundary data. Finally, a single patch is formed by taking a convex combination of the three patches in such a way that the resulting patch interpolates all of the boundary data.

Nielson.

Nielson [26] presented two surface construction techniques. The first is a transfinite method. The input to this scheme is a "triangle" of three boundary curves together with a tangent plane field along each of these curves. The only requirements on each input curve are that it is C^1, and that it meet the other curves with a consistent tangent plane at each vertex. The tangent plane fields are specified using a normal vector field rather than a field of vectors in the tangent plane. That is, at each point along a boundary curve, the tangent plane is the plane perpendicular to the corresponding vector of the normal field. The normal fields are also required to be C^1, with the further restrictions that they must be non-zero everywhere and meet C^0 at the vertices. A surface patch is then constructed that matches this data, using a *side-vertex* method similar to that of the second scheme.

The second scheme is a side-vertex method that fits into the problem domain of this survey. The method proceeds by first constructing three boundary curves, one corresponding to each edge of the input triangle. Three patches are created, one for each boundary/opposite vertex pair. The interior of each patch is constructed by passing curves from points along the boundary (or "side") to the opposite vertex. Hence the name "side-vertex," as shown in Figure 8.5. The three patches are then blended together to form the final patch.

All curves are constructed from two points and associated normals. We assume the existence of a curve construction operator $\mathbf{g}_v$ that takes two vertices with normals and constructs a curve:

$$\mathbf{g}_v[\mathbf{V}_0, \mathbf{V}_1, \vec{\mathbf{N}}_0, \vec{\mathbf{N}}_1](t),$$

such that $\mathbf{g}_v(0) = \mathbf{V}_0$, $\mathbf{g}_v(1) = \mathbf{V}_1$, $\mathbf{g}_v'(0) \cdot \vec{\mathbf{N}}_0 = 0$, and $\mathbf{g}_v'(1) \cdot \vec{\mathbf{N}}_1 = 0$. We will also assume the existence of a normal field constructor $\mathbf{g}_n$ that constructs a continuous normal field along the curve $\mathbf{g}_v$, where $\mathbf{g}_n$ is required to interpolate $\vec{\mathbf{N}}_0$ and $\vec{\mathbf{N}}_1$ at the end points.

The construction proceeds by building three patches, $\mathbf{G}_i$, $i \in \{p, q, r\}$

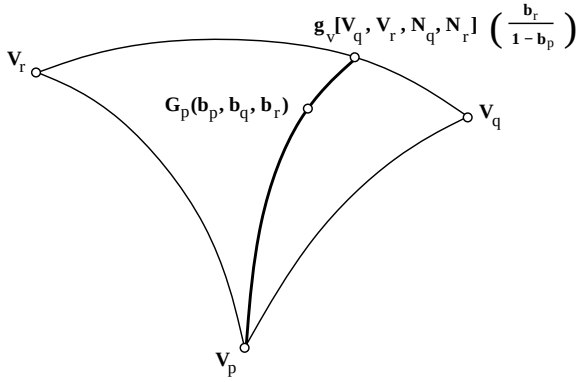

Figure 8.5: Side-vertex method.

defined as:

$$\begin{aligned}\mathbf{G}_i(b_p, b_q, b_r) \quad = \quad & \mathbf{g}_v \left[\mathbf{V}_i,\ \mathbf{g}_v[\mathbf{V}_j, \mathbf{V}_k, \vec{\mathbf{N}}_j, \vec{\mathbf{N}}_k] \left(\frac{b_k}{1-b_i}\right), \right. \\ & \left. \vec{\mathbf{N}}_i,\ \mathbf{g}_n[\mathbf{V}_j, \mathbf{V}_k, \vec{\mathbf{N}}_j, \vec{\mathbf{N}}_k] \left(\frac{b_k}{1-b_i}\right) \right] (1-b_i).\end{aligned}$$

Nielson notes the following two properties of $\mathbf{G}_i$:

1. G_i interpolates all three of the boundaries.
2. G_i interpolates the tangent plane field of the boundary opposite vertex $\mathbf{V}_i$.

The final surface is defined to be

$$\mathbf{G}[\mathbf{V}_p, \mathbf{V}_q, \mathbf{V}_r, \vec{\mathbf{N}}_p, \vec{\mathbf{N}}_q, \vec{\mathbf{N}}_r] = \beta_p \mathbf{G}_p + \beta_q \mathbf{G}_q + \beta_r \mathbf{G}_r,$$

where

$$\beta_i = \frac{b_j b_k}{b_p b_q + b_q b_r + b_r b_p}. \tag{8.8}$$

Nielson shows that the β_i are such that the blending of any three surfaces having the two above properties yields a surface that interpolates all of the boundary curves and tangent fields. The theorem is reasonably general as it is true for a large class of $\mathbf{g}_v$ and $\mathbf{g}_n$. The operator $\mathbf{g}_v$ that Nielson presents constructs tangent vectors from the two normals and interpolates these two points and vectors with a cubic polynomial curve. The construction in

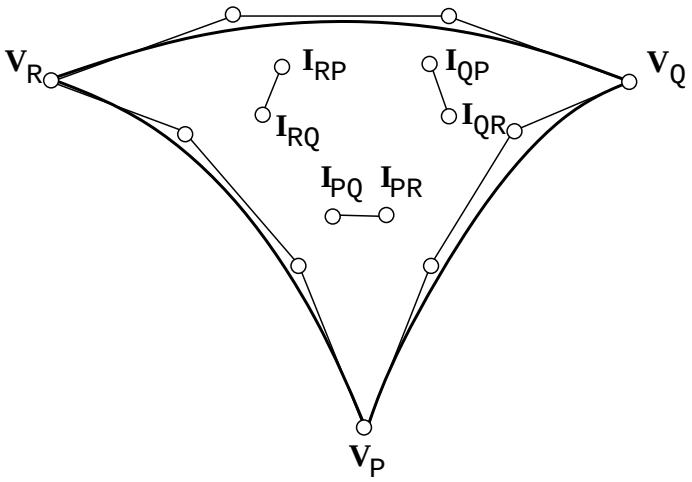

Figure 8.6: Control points of a triangular Gregory patch.

Nielson's paper is not scale invariant, however, since the tangent vectors at the ends of the curve are normalized to unit vectors. This introduces loops in the curves if the data points are close together. The tangents should instead be scaled to be proportional in length to the distance between $\mathbf{V}_0$ and $\mathbf{V}_1$.

Triangular Gregory Patches.

Triangular Gregory patches are a variant of Gregory squares [14]. The key idea is that the twist term at the vertices is a blend of two twists, one for each boundary curve incident to the vertex. The scheme we present here is due to Longhi [22].

After constructing cubic boundary curves, this scheme uses the method of Chiyokura and Kimura to find a pair of cross boundary control points for each edge (Figure 8.6). In this figure, points $\mathbf{I}_{ij}$ and $\mathbf{I}_{ik}$ are the points constructed for the boundary associated with $b_i = 0$. When evaluating the patch at the domain point (b_p, b_q, b_r), the two interior control points near each of the corner vertices are blended to form a single vertex, and thus, the six interior vertices are reduced to three control points. Points $\mathbf{I}_{ij}$ and $\mathbf{I}_{ik}$ are blended to produce the control point $\mathbf{I}_i$ using the following blend:

$$\mathbf{I}_i = \frac{b_k(1-b_j)\mathbf{I}_{ik} + b_j(1-b_k)\mathbf{I}_{ij}}{b_k(1-b_j) + b_j(1-b_k)}.$$

This yields a quartic Bézier patch, which is evaluated at (b_p, b_q, b_r) to give a point on the surface.

Triangular Gregory patches can also be thought of as a convex combination scheme. By putting the blending functions used to construct the $\mathbf{I}_i$'s over a common denominator, the scheme can be rewritten as a convex combination of seven quartic Bézier patches. In this form, the blending functions are sixth degree rational polynomials, four degrees higher than the ones used by Nielson.

Herron.

In [19], Herron introduced a triangular surface fitting scheme in the following form:

$$\mathbf{F} = \mathcal{C} + b_p b_q b_r \mathcal{X},$$

where $\mathcal{C}$ interpolates the boundary curves and $\mathcal{X}$ is presented as a function that adjusts the cross boundary tangents of $\mathcal{C}$ to meet a specified tangent plane fields. Although it can be shown that $\mathbf{F}$ is a point valued function, neither $\mathcal{C}$ nor $\mathcal{X}$ represent affine geometric entities (points, vectors, etc.).

We present here an alternative description of Herron's method that is more geometric in nature. We first observe that $\mathbf{F}$ can be rewritten as

$$\mathbf{F} = \sum_{i=p,q,r} \beta_i \mathbf{F}_i, \tag{8.9}$$

where

$$\beta_i = \frac{b_j b_k}{b_i b_j + b_j b_k + b_k b_i},$$

and where each $\mathbf{F}_i$ is a quartic Bézier patch whose construction is given below. Note that these β_i are identical to those used by Nielson (see Equation 8.8).

The input required by Herron's scheme is a triangle of points and the six boundary curve tangents at those points. Cubic Hermite interpolation is used on the boundary data to construct the boundary curves of $\mathbf{F}_i$. These curves have to be degree raised, as $\mathbf{F}_i$ is a quartic patch. This sets all the exterior control points for $\mathbf{F}_i$, leaving only the three interior control points, $\mathbf{C}_P$, $\mathbf{C}_Q$, and $\mathbf{C}_R$, to be determined (Figure 8.7).

For patch $\mathbf{F}_i$, points $\mathbf{C}_j$ and $\mathbf{C}_k$ are constructed by using the triangular version of Chiyokura-Kimura. Equations 8.4 and 8.5 of Section 8.4.3 give formulas for these two points. The final control point of $\mathbf{F}_i$, $\mathbf{C}_i$, can be

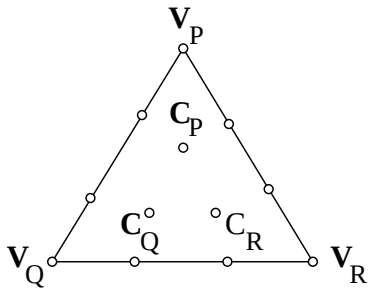

Figure 8.7: Bézier patch for Herron's scheme.

considered a free parameter. Herron's setting for this parameter is given in Appendix 1.

Herron's scheme, then, can be thought of as a hybrid scheme using the cross boundary tangent method of Chiyokura and Kimura for constructing triangular Bézier patches and the weighting functions of Nielson to produce the final patch. Note that we may use Nielson's proof above to show that $\mathbf{F}$ interpolates the tangent plane fields along all edges[3].

A third way to view Herron's scheme is as a "three point" triangular Gregory patch. By rewriting Equation 8.9 in Bernstein form, it is easily seen that Herron's method constructs a quartic Bézier patch, where each internal control point is a blend of three of the internal control points of the $\mathbf{F}_i$s. In Bézier form, then, the three point triangular Gregory patch blends together fewer patches than the two point Gregory patch (three patches instead of seven patches) and uses rational quadratic blending functions instead of the sextic rational polynomials of the two point scheme.

8.5.3 Other Schemes.

There are other parametric surface fitting schemes that fit within the scope of this survey. Among these are one proposed by Farin [8], one proposed by Gregory and Charrot [15], and one proposed by Hagen and Pottmann [18].

Farin's scheme is a split domain scheme that is noteworthy primarily because it was the first parametric triangular surface fitting scheme. The construction, however, has several problems. One problem is that it is asymmetric in its treatment of the neighborhood of control points surrounding a

[3]It is interesting to note that Herron's development of the method predated the publication of Chiyokur and Kimura [4] and Nielson [26]

vertex. This asymmetry is visible in the constructed surfaces, so we chose not to discuss it in detail here.

The method of Gregory and Charrot was originally intended by the authors to be used to fill triangular holes in an array of rectangular tensor product patches. Their scheme is a convex combination scheme that assumes cross boundary tangent fields have already been constructed along the boundary curves. Further, these tangent fields must admit a consistent mixed partial. Extending this scheme to fit into our problem domain would have been a fairly significant change. Although, Gregory [16] later extended this method to allow for inconsistent mixed partials, we realized this too late to include it in this survey.

The discretized interpolant presented by Hagen and Pottmann [18] is another method that falls within our survey. This method extends Nielson's side-vertex method [26] and earlier work by Hagen [17] to second order geometric continuity. It is unfortunate that we learned of this method too late to include it in our survey.

8.6 Comparison.

8.6.1 Tested Characteristics.

Our primary concern in this survey was with the visual appearance of the constructed surfaces. Other concerns, such as computational issues, were considered secondary, as we first wanted to find methods that produced nice shapes. Numerical stability issues are occasionally mentioned, as they can have a large impact on the shape of the resulting surface.

One problem with using visual appearance as our criterion is that it is a subjective measure of surface quality. In part, this stems from a lack of a general purpose "surface quality metric," that is, a commonly agreed upon definition of good shape. However, the problems with the shapes of surfaces we encountered were extreme, leaving little doubt as to the poor quality of the surfaces.

In many applications, the data points themselves are not distinguished points on the surface. Therefore, these points should not be visually distinguishable in the constructed surface. All schemes surveyed in this paper construct piecewise surfaces that have second order derivative discontinuities at the boundaries of the surfaces patches, implying that the boundaries of the patches (and, thus, the data points) will be distinguished to some

Figure 8.8: Line drawing of Clough-Tocher surface.

extent. We feel that the visual impact of these discontinuities should be minimized.

In the past, many authors have used line drawing renditions to show the visual quality of their surfaces. We have found shaded images more useful in detecting various shape defects. For example, the line drawing in Figure 8.8 is a plot of isoparametric lines of the Clough-Tocher interpolant to a function defined as the sum of three Gaussian functions. A shaded image of the same surface (Plate 1) is far more informative. To see more subtle defects we found that Gaussian curvature plots of the interpolants were often helpful. (The Gaussian curvature at a point on a surface is the product of the minimum and maximum normal curvature of the surface at the point.)

All implemented schemes produced surfaces with shape defects that were readily apparent in the shaded images or Gaussian curvature plots. However, a scheme should not be considered "good" just because it passes these two visual tests. Until a good quantitative measure of shape is devised, a surface fitting scheme should also be tested by a variety of other methods, such as reflection lines and isophotes [30], to determine surface quality.

8.6.2 Data Sets.

A variety of data sets were used to test the surface fitting schemes. Several of these are shown in Figures 8.9a-e. In these figures, the lines represent

Figure 8.9: Some of the data sets used: a) A Franke function b) Sphere c) Capsule d) Torus e) Octahedron.

the edges of the triangulated data. The data points are located at the intersection of the lines.

One of the data sets is a sampling of one of the so-called Franke functions [13] (Figure 8.9a). The underlying function is $z = \frac{1}{3}e^{-\frac{81}{4}[(x-.5)^2+(y-.5)^2]}$. The data sets of the sphere and the torus are samplings of those surfaces; the vertices and tangent planes of the octahedron data set have also been sampled from a sphere. The "capsule" data set is a sampling of a truncated cylinder with hemispherical caps.

The sphere data sets tended to be particularly "mild", as the entire surface has positive Gaussian curvature, that is, it has no flat spots or saddle points. The capsule data set was constructed to see if "ringing" would occur along the boundaries between the cylinder and the hemispheres. The torus is probably the most interesting data set, as it has regions of positive, negative, and zero Gaussian curvature. One problem with the dense data sets is that they are somewhat complex, making the resulting surfaces difficult to analyze. The octahedron data set was chosen because it was simple enough to allow us to develop intuition governing the failure of the schemes.

Two materials (i.e., surface reflectance parameters) were used to construct the surfaces appearing in Plates 1-8. The material used in Plates 1,

2, 7, and 8 has a gray diffuse component with a white specular component. The surfaces in Plates 3, 4, 5, and 6 were false shaded to show the Gaussian curvature of the surface. Areas of strongly positive Gaussian curvature are shaded white, with the intensity dropping to a dark gray as the Gaussian curvature goes to zero. Regions of negative Gaussian curvature, which occur at saddle points, were not present in these figures.

8.6.3 Results of Comparison.

The goal of our survey was to find which interpolation schemes produced "nice" surfaces and which schemes did not. We implemented and tested all of the schemes described in Section 8.5 except for the three mentioned in Section 8.5.3. Jensen's and Shirman-Sequin's schemes were similar enough that it seemed adequate to implement only one of them. We chose to implement Shirman and Sequin's scheme using Jensen's generalization of the cross boundary tangents. The software we developed to test these schemes is discussed elsewhere [24]. To our surprise, all of the schemes we tested performed rather poorly. Moreover, they all suffered from shape defects that are qualitatively similar.

Scalar Data.

Initially, we used sparse samplings of some of the Franke functions [13] for our data sets. Running two scalar data schemes on this input, we noticed several problems. First, as has been noted by many others (cf. [11, 12]), we found that the estimation of normals is a difficult problem, and second, these schemes fail to produce nice surfaces on data with high variation.

We decided not to address the problem of normal estimation, choosing to focus instead on performance of the methods once normals had been determined. Even when using normals sampled from a known surface, the shapes of the interpolants were still rather poor. As expected, most of the schemes seemed to perform better on data with less variation, that is, data that was nearly planar.

Parametric Data.

When we started working with parametric data schemes we looked at the interpolants for the data sets shown in Figures 8.9b-e. The shaded images of the interpolants constructed by most schemes for the dense data sets

usually have acceptable visual appearance, but Gaussian curvature plots indicate that there are subtle problems with them.

For example, inspection of Gaussian curvature plots for interpolants to the sphere data reveals that the patches are mostly flat, with a few areas of high curvature (Plate 5). These effects occurred fairly uniformly for most schemes, with additional curvature discontinuities appearing along the interior boundaries produced by split domain schemes that are not present in the convex combination schemes. As a representative scheme, we show pictures of the surfaces constructed by Shirman and Sequin's scheme.

The one scheme that had additional kinds of difficulties to those problems mentioned above was Piper's scheme. Surfaces constructed by this scheme often exhibit displeasing undulations near the patch boundaries. This appears to be a result of numerical instabilities, occurring when the scalar functions of Equations 8.2 are nearly constant (and, thus, when the 3×5 system of equations nearly reduces to a 2×5 system). We decided not to address these numerical stability issues, focusing instead on the other schemes.

Shaded images of the interpolants to the torus data revealed problems more serious than the ones mentioned above (Plate 7). Images of Gaussian curvature indicated that there were large variations of curvature, even within a single patch. Discontinuous jumps from positive to negative curvature were also observed along patch boundaries.

Images of Gaussian curvature for interpolants to the simple data set of the octahedron (Plate 2) clearly show that curvature is concentrated near the vertices and boundary curves (Plate 3). The center of the patch (or patches) created for a face tend to be relatively flat by comparison. These problems were similar for all the schemes, which was somewhat unexpected, considering that the split domain schemes construct surfaces in a very different fashion from the convex combination schemes.

Some of these problems could be alleviated by manually adjusting the scalar shape parameter of Jensen's scheme. For example, the curvature of the surfaces constructed for the octahedron could be spread over the patches somewhat more uniformly, but there are still flat spots on these surfaces. The improvements possible for the torus data set are less significant. It is also unclear how to set automatically the value of this parameter.

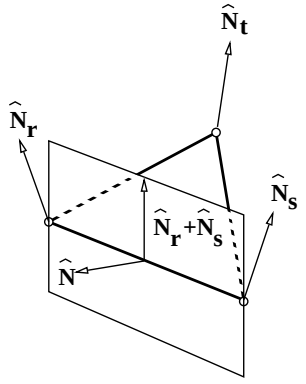

Figure 8.10: The construction of the tangent at $\mathbf{P}$ for the curve from $\mathbf{P}$ to $\mathbf{Q}$. $|\vec{\mathbf{T}}| = |\mathbf{P} - \mathbf{Q}|$.

8.6.4 Boundary Curves.

The construction of boundary curves was a common element in all our implementations. Following constructions given by Piper and Shirman-Sequin, we originally used the following method for constructing a cubic boundary curve for an edge of the data set: The end points of the curve are set to interpolate the end points of the edge. Next, tangent directions are computed for each end point by perpendicularly projecting the vector along the edge into the tangent plane at each end point; these vectors are then scaled to be of length equal to the length of the edge (see Figure 8.10). The boundary curve is then set to be the cubic polynomial curve matching this data.

As published, the surveyed schemes used a variety of methods for constructing boundary curves. However, the differences between these methods are minor. Shirman and Sequin use the method mentioned above. Herron and Jensen essentially assume that the boundary curves have already been constructed. Nielson constructs cubic boundary curves whose tangents agree in direction with the tangents in the above construction, but leaves the length as a free parameter. Piper's scheme is unique among these schemes in that it constructs quartic boundary curves; the end point tangents have the same direction as those in the method presented above, but are shorter in length.

Curvature plots of boundary curves constructed as above showed that as the tangent vectors become more perpendicular to the vector along the edge the curvature along the curve is concentrated near the end points (Fig-

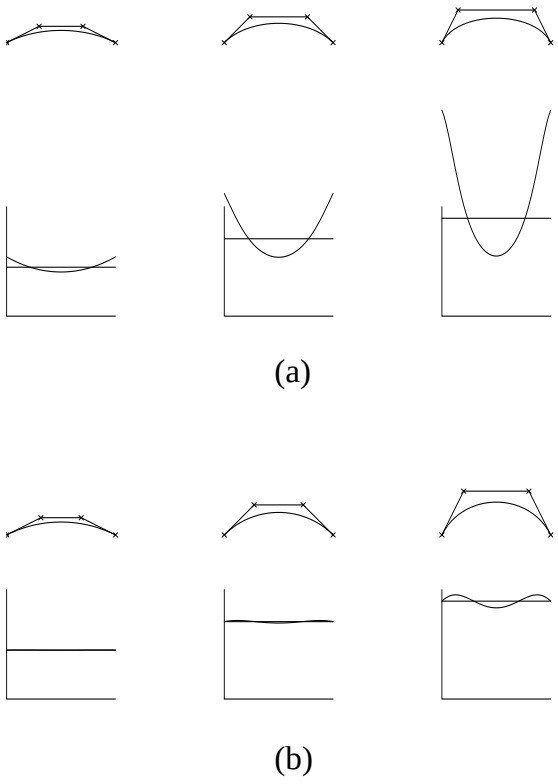

Figure 8.11: Bézier curves and their curvature plots. (a) Curves constructed using projection method. (b) Curve constructed by de Boor-Höllig-Sabin method.

ure 8.11a), leaving a relatively flat region in the middle of the curve. In all the surface schemes we implemented, this flatness was then propagated inward in the patch construction, resulting in relatively large flat areas in the middle of the patches. This suggested that it might be possible to improve the curvature distribution of the patches by improving the curvature distribution of the boundary curves.

In a paper by de Boor, Höllig, and Sabin [5], a method is given for constructing planar cubic curves that interpolate to positions, tangent lines, and curvatures at two end points. More importantly for our purposes, these curves were observed to have relatively uniform curvature distributions. In Figure 8.11b, for example, the curvature values are taken from the circle passing through the data points with the given tangents. If the curvature

values at the end points had not been equal, the resulting curves would have been approximations to an ellipse.

To determine if using boundary curves with a more uniform distribution of curvature would yield surfaces with better shape, the de Boor-Höllig-Sabin curve construction technique was integrated into each of the surface fitting schemes. The resulting interpolants all show improvement in shape, with the interpolants to the simpler data sets showing more improvement than the interpolants to more complex ones. Interpolants to the sphere data set, for example, exhibit nearly uniform distribution of Gaussian curvature (Plate 6).

The results for the octahedron data set are not as encouraging. The curvature is spread along the patches somewhat more uniformly, but the patches are still flat in the interior (Plate 4). Additional improvements can be made by manually adjusting Jensen's shape parameter.

The improvement in the shape of the interpolants for the torus data set are minor. Using the de Boor-Höllig-Sabin method yields an interpolant that is a better approximation to the torus (using a radial distance metric); however, while the shape defects are alleviated somewhat, the shaded image of the surface still shows many of the shape defects apparent in the surface produced by the standard boundary curve method (Plates 7 and 8). So, while the de Boor-Höllig-Sabin boundary curve method appears to improve the shape of the interpolants, it is not a complete solution. There are also several problems with using this method for constructing boundary curves.

First, in order to have the curvature information needed by the de Boor-Höllig-Sabin scheme, second fundamental forms (cf. doCarmo [6]) must be associated with the vertices of the data sets. If the data are sampled from a C^2 function, then second fundamental forms can be calculated directly. If only the data points are available, then second fundamental forms must be estimated. It is not clear at this time how difficult it is to make these estimates.

A more serious problem is that there are from zero to three cubic curves which match the curvature data [5]. For our purposes, if there are zero curves that interpolate the data, then the boundary curves must be computed by some other method. If there are multiple solutions, then a choice must be made between these solutions.

The case of multiple solutions revealed another interesting fact. In the cases we saw, all of the solution curves given by the de Boor-Höllig-Sabin method have essentially the same shape. These solutions differ primarily in their parametrizations. For the sphere data, for example, there are three

distinct cubic curves that match the end point data. The solution that is most nearly uniform in parameterization gives the surface shown in Plate 6. However, when one of the less uniform parameterizations is used, large lumps appear in the surface. Thus, both the shape and the parameterization of the boundary curves are important.

Another issue concerning the use of the de Boor-Höllig-Sabin method for constructing boundary curves is that it mandates the use of planar curves. Thus, a plane must be chosen in which to place each boundary curve. The choice of this plane can be thought of as a free parameter. The plane we used is the one containing the edge, and whose normal vector is the cross product of the vector along the edge and the average of the normals at the end points. It is unclear how restricting the boundary curves to lie in a plane affects the shape of the surfaces.

8.7 Summary and Recommendations.

At the beginning of our study we did not expect to find a method that would work well for arbitrarily placed data. It soon became clear, however, that we had drastically underestimated the difficulty of the problem. As expected, the schemes produced poor interpolants to extremely sparse data sets. More surprising was how hard it was to produce good interpolants for any but the most benign data sets. In particular, we observed the following:

- Although different schemes constructed surfaces in different fashions, all surfaces displayed similar shape defects.
- The primary cause of the shape defects appears to be in the construction of boundary curves. Flatness on the boundary curves is propagated inward, resulting in flat spots on the surface.

The shape and parameterization of the boundary curves greatly influences the shape of the patches. To construct surfaces with better shape, a curve construction method must be found that spreads the curvature uniformly along the boundary curves. One way to construct such curves might be to relax the locality condition on the schemes and do some form of global optimization on the boundary curves, perhaps something similar to Nielson's minimum norm networks [25, 27]. Alternatively, a local method based on curvature interpolation and de Boor-Höllig-Sabin's method seems promising, at least for approximating known surfaces. A disadvantage of such an approach is that good methods of estimating second fundamental forms would

need to be developed. It is also unclear what ramifications the restrictions to planar boundary curves might have.

Improving the shape of boundary curves, however, is not a complete solution to the construction of surfaces free of unnecessary shape defects. Many schemes provide additional shape parameters that can be adjusted to improve the appearance of the interpolant. Such shape parameters may be useful in interactive design applications, but it is important to develop methods for setting good default values. In the approximation of known surfaces, it is also important to understand how differential geometric properties of the known surface can be used to set the free parameters.

8.7.1 Recommendations.

Since all of the schemes produce surfaces with similar shapes, we recommend using the Shirman-Sequin scheme, but only because it constructs polynomial patches rather than the rational polynomial patches built by most of the other schemes. The use of polynomial patches simplifies the calculation of derivatives, curvature, etc. of the interpolant. Although the domain split introduces extra artifacts in the surfaces, such as the creation of long, thin triangles, the major shape defects are common to all schemes.

Triangular Gregory patches are often suggested as the scheme of choice. However, it is hard to determine differential information since the patches are rational polynomials of fairly high degree. Further, we found that this scheme is extremely sensitive to the settings of the free parameters. While this might be desirable in some situations, it appears to be difficult to control these shape parameters and to determine reasonable default values for them.

Estimating second fundamental forms at the data points appears to have two benefits. First, the second fundamental form can be used to solve the vertex consistency problem. It can also be used to produce boundary curves with more uniform curvature, resulting in surfaces with better shape. When approximating known surfaces, the surface can be sampled for position, tangent, and second fundamental form (assuming it is C^2). Using all of this information in the construction of the approximating surface, a higher order of convergence should be achievable.

8.8 Acknowledgements.

This work was supported in part by the National Science Foundation under grant numbers DMC-8802949, CCR-8957323, CCR-8612543, and IRI-

8801932. Support from the Digital Equipment Corporation, Xerox, and IBM is also gratefully acknowledged.

Appendix 1

This appendix gives Herron's setting of the control point $\mathbf{C}_i$ (Figure 8.7). We will denote the tangent from point $\mathbf{V}_i$ to $\mathbf{V}_j$ as $\vec{\mathbf{T}}_{ij}$. In the construction of patch $\mathbf{F}_i$, Herron implicitly sets $\mathbf{C}_i$ to be:

$$\begin{aligned}\mathbf{C}_i &= \tfrac{1}{12}\left\{-6\mathbf{V}_j - 2\vec{\mathbf{T}}_{ji} + \vec{\mathbf{T}}_{ik} - \vec{\mathbf{T}}_{ij} + \tfrac{9}{4}\left[\vec{\mathbf{R}}(\tfrac{1}{3}) - \mathcal{Q}(\tfrac{1}{3})\right]\right. \\ &\quad \left. + \tfrac{9}{4}\left[\vec{\mathbf{R}}(\tfrac{2}{3}) - \mathcal{Q}(\tfrac{2}{3})\right]\right\} + \mathbf{E}^4_{i2},\end{aligned}$$

where

$$\mathbf{E}^4_{i2} = \frac{1}{2}\mathbf{V}_j + \frac{1}{6}\vec{\mathbf{T}}_{ji} + \frac{1}{2}\mathbf{V}_i + \frac{1}{6}\vec{\mathbf{T}}_{ij},$$

and $\vec{\mathbf{R}}(t) = k(t)\cdot\vec{\mathbf{C}}(t)$ (from Section 8.4.3) and

$$\begin{aligned}\mathcal{Q}(t) &= 6t(1-t)p_j(t)\cdot\mathbf{V}_j + 6t(1-t)p_k(t)\cdot\mathbf{V}_k \\ &\quad + [(1-t)^2 p_k(t) + 2t(1-t)p_j(t)]\cdot\vec{\mathbf{T}}_{jk} \\ &\quad + [t^2 p_j(t) + 2t(1-t)p_k(t)]\cdot\vec{\mathbf{T}}_{kj} \\ &\quad + (1-t)^2\vec{\mathbf{T}}_{ji} + t^2\vec{\mathbf{T}}_{ki}.\end{aligned}$$

Here $p_j(t)$ and $p_k(t)$ are the scalar functions

$$p_j(t) = t(r_{ji} + r_{ki} + 1) - r_{ji} - 1$$

and

$$p_k(t) = (1-t)(r_{ji} + r_{ki} + 1) - r_{ki} - 1,$$

where r_{ji} and r_{ki} are:

$$r_{ji} = \frac{\vec{\mathbf{T}}_{jk}\cdot\vec{\mathbf{T}}_{ji}}{\vec{\mathbf{T}}_{jk}\cdot\vec{\mathbf{T}}_{jk}} \quad \text{and} \quad r_{ki} = \frac{\vec{\mathbf{T}}_{kj}\cdot\vec{\mathbf{T}}_{ki}}{\vec{\mathbf{T}}_{kj}\cdot\vec{\mathbf{T}}_{kj}}.$$

Bibliography

[1] R. Barnhill. Computer Aided Surface Representation and Design. In R. Barnhill and W. Boehm, editors, *Surfaces in computer aided geometric design*, pages 1–24. North-Holland, 1983.

[2] W. Boehm. Visual continuity. *Computer Aided Design*, 20(6):307–311, 1988.

[3] H. Chiyokura. Localized surface interpolation method for irregular meshes. In Tosiyasu L. Kunii, editor, *Advanced Computer Graphics*, pages 3–19. Springer-Verlag, 1986.

[4] H. Chiyokura and F. Kimura. Design of solids with free-form surfaces. *Computer Graphics*, 17(3):289–298, 1983.

[5] C. de Boor, K. Höllig, and M. Sabin. High accuracy geometric Hermite interpolation. *CAGD*, 4(4):269–278, December 1987.

[6] M. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice Hall, 1976.

[7] G. Farin. A construction for visual C^1 continuity of polynomial surface patches. *Computer Graphics and Image Processing*, (20):272–282, 1982.

[8] G. Farin. Smooth interpolation to scattered 3D data. In R. Barnhill and W. Boehm, editors, *Surfaces in Computer Aided Geometric Design*, pages 43–63. North-Holland, 1983.

[9] G. Farin. A modified Clough-Tocher interpolant. *CAGD*, 2(1-3):19–27, September 1985.

[10] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design.* Academic Press, 1988.

[11] R. Franke. Scattered data interpolation: tests of some methods. *Mathematics of Computation*, 38(157):181–200, January 1982.

[12] R. Franke and G. Nielson. Scattered data interpolation: A tutorial and survey. In H. Hagen, editor, *Geometric Modeling: Methods and Applications*. Springer-Verlag, to appear 1990.

[13] T. Grandine. An iterative method for computing multivariate C^1 piecewise polynomial interpolants. *CAGD*, 4(4):307–319, December 1987.

[14] J. Gregory. Smooth interpolation without twist constraints. In R. Barnhill and R. Riesenfeld, editors, *Computer Aided Geometric Design*, pages 71–87. Academic Press, 1974.

[15] J. Gregory and P. Charrot. A C^1 triangular interpolation patch for computer-aided geometric design. *Computer Graphics and Image Processing*, 13:80–87, 1980.

[16] J. Gregory. N-sided surface patches. In J. Gregory, editor, *The Mathematics of Surfaces*, pages 217–232. Clarendon Press, 1986.

[17] H. Hagen. Geometric surface patches without twist constraints. *CAGD*, 3(3):179-184, November 1986.

[18] H. Hagen and H. Pottmann. Curvature continuous triangular interpolants. In T. Lyche and L. L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design*, pages 373–384. Academic Press, 1989.

[19] G. Herron. Smooth closed surfaces with discrete triangular interpolants. *CAGD*, 2(4):297–306, December 1985.

[20] G. Herron. Techniques for visual continuity. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 163–174. SIAM, 1987.

[21] T. Jensen. Assembling triangular and rectangular patches and multivariate splines. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 203–220. SIAM, 1987.

[22] L. Longhi. Interpolating patches between cubic boundaries. Technical Report T.R. UCB/CSD 87/313, University of California, Berkeley, Berkeley, CA 94720, October 1986.

[23] C. Loop. A G^1 triangular spline surface of arbitrary topology. In preparation.

[24] M. Lounsbery, C. Loop, S. Mann, D. Meyers, J. Painter, T. DeRose, and K. Sloan. A testbed for the comparison of parametric surface methods. In L. A. Ferrari and R. J. P. de Figueiredo, editors, *Curves and Surfaces in Computer Vision and Graphics,* pages 94-105, SPIE Proceedings Volume 1251, February, 1990

[25] G. M. Nielson. Minimum norm interpolation in triangles. *SIAM Journal of Numerical Analysis*, 17(1):44–62, February 1980.

[26] G. M. Nielson. A transfinite, visually continuous, triangular interpolant. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 235–246. SIAM, 1987.

[27] G. M. Nielson. Iteractive surface design using triangular network splines. In *International Conference on Engineering Graphics and Descriptive Geometry Proceedings*, volume 2, pages 70–77. 1988.

[28] J. Peters. A local interpolation of a cubic curver mesh by a piecewise [bi]quartic C^1 surface without splitting. To appear in Constructive Approximation.

[29] B. Piper. Visually smooth interpolation with triangular Bézier patches. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 221–233. SIAM, 1987.

[30] T. Poeschl. Detecting surface irregularities using isophotes. *Computer Aided Geometric Design*, 1(2):163–168, 1984.

[31] L. A. Shirman and C. H. Séquin. Local surface interpolation with Bézier patches. *CAGD*, 4(4):279–295, December 1987.

[32] J. J. van Wijk. Bicubic patches for approximating non-rectangular control-point meshes. *CAGD*, 3(1):1–13, May 1986.

Plate 1. Clough-Tocher interpolant constructed using estimated normals.

Plate 2. Shirman-Sequin interpolant constructed for octahedron data set.

Plate 3. Gaussian curvature plot of the Shirman-Sequin interpolant constructed for octahedron data set (Plate 2).

Plate 4. Gaussian curvature plot of the interpolant constructed for octahedron data set using de Boor-Höllig-Sabin method for computing boundary curves.

Plate 5. Gaussian curvature plot of the Shirman-Sequin interpolant constructed for sphere data set.

Plate 6. Gaussian curvature plot of the interpolant constructed for sphere data set using de Boor-Höllig-Sabin method for computing boundary curves.

Plate 7. Shirman-Sequin interpolant constructed for torus data set.

Plate 8. Interpolant constructed for torus data set using de Boor-Höllig-Sabin method for computing boundary curves.