A Parametric Hybrid Triangular Bézier Patch

by

Matthew Francis Davidchuk

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 1997

I hereby declare that I am the sole author of this thesis.

I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Waterloo to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

The triangular Bézier patch presents a natural primitive for modeling surfaces of arbitrary topology, but does not enjoy the same widespread use as the tensor product patch. Existing triangular Bézier patch interpolation schemes that are designed to fit surfaces to *parametric data* produce interpolants that suffer from noticeable visual flaws. Patch boundaries are often visible as creases in the resulting surface.

Interpolation of functional data is simpler than interpolation of parametric data. However schemes such as the Clough-Tocher technique have many of the same visual flaws as parametric schemes. Foley and Opitz introduced an improvement over Clough-Tocher. Central to the Foley-Opitz scheme is the cross boundary construction that produces quadratically varying cross boundary derivatives.

My work involves attempting to extend the Foley-Opitz scheme to the parametric setting. Modifying this scheme requires the formulation of an extension to the standard rational blend technique. In addition to blending interior control points, boundary control points must also be blended to incorporate Foley-Opitz cross boundary construction, which relies on the natural parameterization of the functional setting.

When interpolating irregularly scattered data and when increasing the tessellation of the data mesh, the new scheme shows improvement over representative parametric data fitting schemes. The quality of the resulting surfaces depended largely on the correlation between the normals and the triangles of the underlying mesh.

Acknowledgments

I would like to acknowledge the Information Technology Research Institute for providing me with financial support as a masters student at the University of Waterloo.

I would also like to thank Stephen Mann, my supervisor, for his guidance, patience and insight. I also appreciate the corrections and suggestions from my faculty readers, Richard Bartels and Hans-Peter Seidel.

Contents

List of Figures

Chapter 1

Introduction

Modeling in computer graphics deals with the mathematical description of objects in a scene. Surfaces can be modeled with polygonal meshes, which offer simplicity at the cost of large memory requirement and difficult mesh manipulation. Polynomial spline surfaces solve both of these problems by using a relatively sparse set of control points to define a surface that can later be dynamically tessellated into a polygonal mesh. Mathematical conditions on the placement of control points that ensure smooth joins between splines, also known as patches, allow higher level manipulation of surfaces – instead of worrying about many polygons while deforming a surface, relatively few control points of a spline need be modified. Extra control points that ensure smooth joins can then be inserted automatically.

Fitting data supplied by the user of a surface modeling system requires points and, potentially, normals in $\Re^3$. The resulting spline, $\mathbf{F}$, will have the form

$$\mathbf{F}(s,t) = \begin{bmatrix} x(s,t) \\ y(s,t) \\ z(s,t) \end{bmatrix}.$$

and will interpolate the data or provide a good approximation. This type of spline and data are said to be *parametric.*

The tensor product patch, specifically the B-spline tensor product patch, is the spline primitive of choice for constructing surfaces because of its high order continuity, local control, and interactive nature. The triangular spline is another primitive, but it has not enjoyed the same widespread use as the tensor product B-spline even though it dates back to the birth of computer aided geometric design. Triangular splines are interesting and potentially very powerful because they can naturally describe surfaces of arbitrary topology. The tensor product patch maps a rectangular domain to an affine space, giving it an inherently rectangular nature. However, constructing surfaces that are not topologically equivalent to the plane may present some problems. Triangular holes may have to be filled by constructing degenerate tensor product patches – two boundaries are collapsed into one boundary to fit the triangular hole.

The triangular Bézier patch is a two dimensional extension of the Bézier curve that preserves several of the properties of the one dimensional case. Unfortunately, existing triangular patch interpolation schemes have a number of drawbacks and must balance surface quality with interactivity. Schemes that provide efficient evaluation, and hence interactivity, produce surfaces of visually poor quality. Most schemes that improve on this quality use some type of global or large scale optimization that makes interactivity infeasible. Triangular parametric patches have the problem that higher degree patches must be used to fit the data, but the patch formulation leaves many of the degrees of freedom of the high degree patch unused. These degrees of freedom are then left as *shape parameters* for a modeler to adjust or are set using some heuristic. Global optimization schemes attempt to set these leftover parameters by minimizing some metric. A third approach is to use DMS splines [3], which have high order continuity, but unfortunately are expensive to evaluate.

Mann [16] conducted a survey of local triangular spline interpolation schemes and has concluded that none of the existing schemes produce high quality surfaces because of poor boundary curve construction. I will consider the problem of improving existing local, parametric interpolation schemes that use triangular Bézier patches, or variants that are known as hybrid triangular Bézier patches.

One setting in which triangular Bézier patches currently work well is the fitting of *functional* data with *functional* patches. Functional patches are parameterized by the plane and hence represent a restricted subset of general parametric patches. They have the form

$$\mathbf{G}(s,t) = \begin{bmatrix} s \\ t \\ z(s,t) \end{bmatrix}.$$

The good results given by functional patches are a result of the restricted nature of the functional setting – for cubic patches boundary curves are completely determined by the triangle vertices and normals. This is in contrast to the parametric setting in which cubic curve construction has four degrees of freedom.

My approach is to extend a good functional interpolation scheme, the scheme of Foley and Opitz [10], to the parametric setting. In Chapter 2, I present some background material about triangular Bézier patches, parametric schemes, and functional schemes. In Chapter 3, I consider the two elements of patch construction, boundary curve and cross boundary construction, and present the Foley-Opitz scheme. In Chapter 4, I formulate a parametric version of the Foley-Opitz scheme that includes the formulation of a new hybrid Bézier patch. In Chapter 5, I analyze the new scheme by constructing interpolants for some test data. Comparisons with other parametric schemes are made, plus an analysis of why the new scheme gives some superior results. In Chapter 6, I present my conclusions, make some recommendations about using the new patch, and point to some future work that could be done in this area.

Chapter 2

Background

In this chapter, I present some background material on triangular Bézier patches, explain the problem of fitting scattered parametric data, and discuss functional data fitting.

2.1 Triangular Bézier Patches

The triangular Bézier patch is a simple primitive that can be used to interpolate scattered data while offering interactive manipulation and local control of a surface. There is no inherent relationship between the domain and range of triangular Bézier patches, so they are candidates for interpolating scattered parametric data, i.e., producing a function of the form

$$\mathbf{F}(s,t) = \begin{bmatrix} x(s,t) \\ y(s,t) \\ z(s,t) \end{bmatrix}$$

that interpolates scattered points in $\Re^3$.

2.1.1 Bernstein Polynomials

The m variate, homogeneous Bernstein polynomials, or Bernstein basis functions, of degree n are

$$B^n_{\vec{i}}(t_0, t_1, \ldots, t_m) = \begin{pmatrix} n \\ \vec{i} \end{pmatrix} t_0^{i_0} t_1^{i_1} \cdots t_m^{i_m} \tag{2.1}$$

where

$$\vec{i} = (i_0, \ldots, i_m),\ i_j \geq 0,\ \sum i_j = n$$

and

$$\begin{pmatrix} n \\ \vec{i} \end{pmatrix} = \frac{n!}{i_0! i_1! \cdots i_m!}.$$

They are essential building blocks in the formulation of Bézier curves and surfaces.

2.1.2 Bézier Curves

A degree n Bézier curve

$$\mathbf{F}(t) = \sum_{i=0}^{n} B_i^n(t) \mathbf{P_i}$$

uses the *control points* $\mathbf{P_i}$ and the univariate Bernstein polynomials to map a segment of the real line $[a, b]$ to an affine space, typically $\Re^2$ or $\Re^3$.

Among the useful properties of Bézier curves are

1. End point interpolation – $\mathbf{F}(a) = \mathbf{P_0}$ and $\mathbf{F}(b) = \mathbf{P_n}$ so the Bézier curve contains the first and last control points, a useful property for data fitting.

2. Convex hull property – For $t \in [a, b]$ the curve lies in the convex hull of the control points $\mathbf{P_i}$.

2.1.3 Triangular Bézier Patches

Triangular Bézier patches are natural extensions of Bézier curves to two dimensions. Using a triangular network of control points

$$\mathbf{P}_{\vec{i}} : \vec{i} = (i_0, i_1, i_2),\ i_0 + i_1 + i_2 = n,\ i_j \geq 0, i_j \text{ are integers}$$

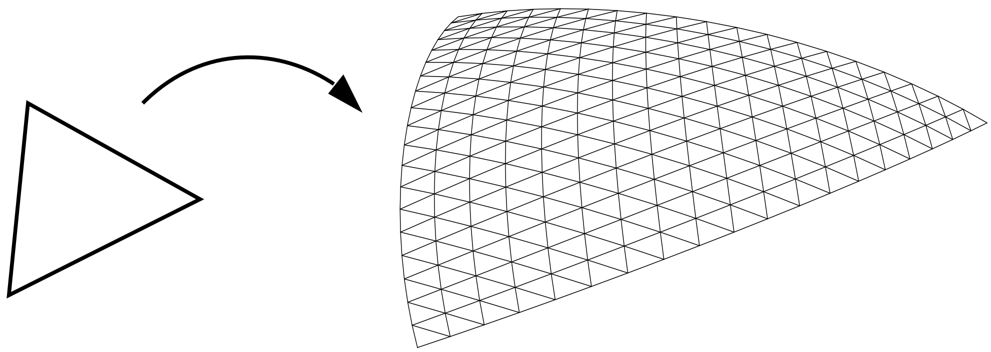

Figure 2.1: Triangular Bézier patches are polynomials mapping triangular regions of the real plane into an affine space.

and the trivariate Bernstein polynomials

$$B^n_{\vec{i}}(t_0, t_1, t_2) = \begin{pmatrix} n \\ \vec{i} \end{pmatrix} t_0^{i_0} t_1^{i_1} t_2^{i_2},$$

a degree n triangular Bézier patch

$$\mathbf{F}(t_0, t_1, t_2) = \sum_{\vec{i}} \mathbf{P}_{\vec{i}} B^n_{\vec{i}}(t_0, t_1, t_2), \tag{2.2}$$

maps a triangular domain, $D \in \Re^2$, to an affine space, typically $\Re^3$, where t_0, t_1, and t_2 are the barycentric coordinates of a domain point relative to D.

When considered over the domain, $D \in \Re^2$, Bézier patches have some interesting properties:

1. Corner point interpolation – the two dimensional analogue of end point interpolation. The Bézier patch contains the three corner control points but in general does not contain any other control points.

2. Convex hull property – The image of the triangular domain, $D \in \Re^2$, lies in the convex hull of the control points.

3. Bézier boundary curves – The images of the three edges of the domain triangle are Bézier curves defined by the boundary control points.

2.2 Scattered Data Fitting

The problem of scattered data fitting can be described as the problem of constructing a function to interpolate a set of sparse data. I will only be considering the fitting of points in $\Re^3$ and associated normals in this discussion.

"Scattered" means that the data is sufficiently sparse that interpolation will produce some meaningful and efficient representation and that each piece of data contributes significantly to the local structure of the interpolant. On the other hand, fitting dense data requires some representation that indicates the overall structure of the data but gives little information about the local structure of the data, i.e., an approximation. "Scattered" also implies a non-uniform distribution of data. The data is assumed to have some structure – points are organized into a mesh so that "neighbours" of each point are known. Issues such as mesh construction from unorganized data or estimation of normals from data points, are beyond the scope of this discussion.

2.2.1 Why Use Triangular Patches?

Two types of surfaces that can be used to solve the scattered data fitting problem are the triangular patch and the tensor product patch. The underlying geometry of the triangular patch is a triangle. The underlying geometry of the tensor product is the quadrilateral.

Bézier and B-spline tensor product patches have simple cross boundary constructions, quick evaluation and high order continuity with local control, all highly desirable properties for surface patches. However, the underlying geometry of tensor product patches presents problems when modeling complex topologies.

Triangular Bézier patches have been around since the birth of computer aided geometric design – de Casteljau investigated them as extensions of Bézier curves to surfaces. They are a natural choice for data fitting and surface construction since surfaces of arbitrary topology can easily be decomposed into triangles. Unfortunately, triangular Bézier patches have not been developed to the same level as tensor product patches and consequently are missing some of the features of tensor product B-splines, most noticeably high order continuity. Existing triangular Bézier patch data fitting techniques produce noticeable visual defects, demonstrating the lack of high order continuity.

Other promising interpolation techniques, such as triangular B-splines [3], offer features by using primitives more complex than the triangular Bézier patch. The increased complexity unfortunately results in expensive evaluations, difficulty of use for the modeler, and thorny implementation issues.

The Bézier patch is a simple primitive that not only interpolates data but facilitates interactivity, local control, and natural construction of surfaces of arbitrary topology.

2.2.2 Continuity

To be useful for surface design a data fitting scheme must produce a continuous surface that is smooth. I will outline continuity for parametric patches and present one G^1 construction, Chiyokura-Kimura's construction, that is an essential component of two of the interpolation schemes discussed later.

Zeroth Order Continuity

The boundary Bézier curve of a triangular Bézier patch is completely determined by the boundary control points so if two patches share the common boundary control points they will meet with C^0 continuity.

First Order Continuity

From a geometric perspective, the mathematical term C^1 does not characterize the smoothness of a surface well. For example, two neighbouring patches may share a border and cross boundary derivatives along that border, but a change in the parameterization of one of the patches will change the cross boundary derivatives of that patch making them no longer joined with C^1 continuity. There is no natural way to associate the domains of parametric patches.

Tangent plane continuity, also known as G^1 continuity, is a better definition of the first order continuity of a surface. Two patches are said to be tangent plane continuous, or G^1 continuous if they have C^0 continuity along one boundary and have matching unit normals along that boundary. See Farin's book [5] for a formal description of G^1 continuity. Farin has published a paper outlining simple conditions that are sufficient for two triangular Bézier patches to meet with G^1 continuity [7].

Chiyokura-Kimura

Of the various approaches to constructing surface patches, one technique is to first construct a field of cross boundary tangent vectors using data from two neighbouring patches. Chiyokura and Kimura [1] developed a technique that constructs a G^1 join between two bicubic tensor product Bézier patches. The method can be extended to quartic triangular patches with cubic boundary curves.

The following diagram shows the boundary separating two quartic patches plus the control points used in the construction. Note that the four boundary control points define a cubic curve but the two patches are quartic. A cubic curve is used so that the G^1 construction can exactly mirror the construction for the case of a bicubic tensor product patch. The curve will be degree raised to a quartic after construction of the tangent plane field. Let $\mathbf{F}$ be the patch with control points $\mathbf{F_0}, \mathbf{F_1}, \mathbf{F_2}, \mathbf{F_3}, \mathbf{H_0}, \mathbf{H_1}, \mathbf{H_2}$, and $\mathbf{H_3}$, and let $\mathbf{G}$ be the patch with control points $\mathbf{G_0}, \mathbf{G_1}, \mathbf{G_2}, \mathbf{G_3}, \mathbf{H_0}, \mathbf{H_1}, \mathbf{H_2}$, and $\mathbf{H_3}$,.

Any technique can be used to construct $\mathbf{H(t)}$, the cubic boundary curve. For patch $\mathbf{F}$, unit cross boundary tangent vectors, $\mathbf{\hat{C}_0}$ and $\mathbf{\hat{C}_1}$, are chosen at the two patch vertices $\mathbf{H_0}$ and $\mathbf{H_1}$. The cross boundary tangent field for $\mathbf{F}$ is chosen to be the linear blend

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

$\vec{\mathbf{C}}(t)$ and $\mathbf{H'(t)}$ completely define the tangent plane field along the boundary. Note that when working with patch $\mathbf{G}$, $-\mathbf{\hat{C}_0}$ and $-\mathbf{\hat{C}_1}$ would be used as cross boundary tangent vectors at the patch vertices.

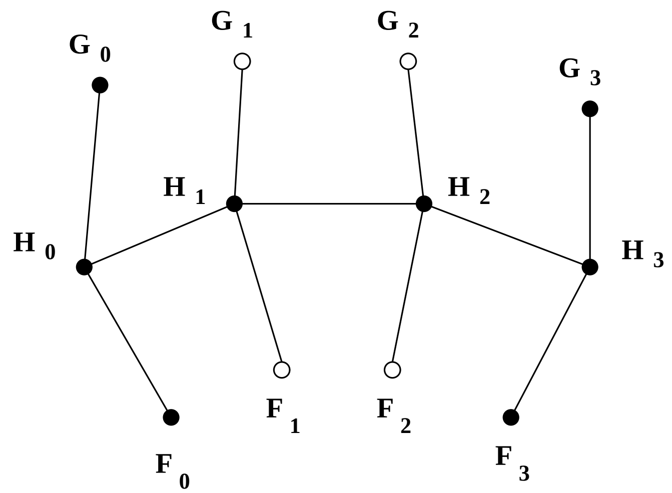

Figure 2.2: Boundary between two quartic patches.

For patch $\mathbf{F}$ to agree with this tangent plane field there must exist a function

$$\mathbf{D}_{\vec{\mathbf{r}}(\mathbf{t})}\mathbf{F}(0,t,1-t) = k(t)\cdot\vec{\mathbf{C}}(t) + h(t)\cdot\mathbf{H}'(t) \tag{2.3}$$

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

$k(t)$ and $h(t)$ can be evaluated at $t=0$ and $t=1$ by using Equation 2.3.

$$\vec{\mathbf{F}}_{\mathbf{0}} = k_0\cdot\hat{\mathbf{C}}_{\mathbf{0}} + h_0\cdot\vec{\mathbf{H}}_{\mathbf{0}}$$

$$\vec{\mathbf{F}}_{\mathbf{3}} = k_1\cdot\hat{\mathbf{C}}_{\mathbf{1}} + h_1\cdot\vec{\mathbf{H}}_{\mathbf{1}}$$

where $\vec{\mathbf{H}}_{\mathbf{i}} = \mathbf{H}_{\mathbf{i+1}} - \mathbf{H}_{\mathbf{i}}, \vec{\mathbf{H}}_{\mathbf{i}} = \mathbf{H}_{\mathbf{i+1}} - \mathbf{H}_{\mathbf{i}}, k_0 = k(0), k_1 = k(1), h_0 = h(0)$, and $h_1 = h(1)$. Since, in general, $k_0 \neq k_1$ and $h_0 \neq h_1$ both $h(t)$ and $k(t)$ must be at least linear functions. The construction is made easier if both h and k are restricted to be linear functions.

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

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

Equation 2.3 can be expressed in terms of the cubic Bernstein basis functions. The coefficients for $B_1^3(t)$ and $B_2^3(t)$ can then be used to determine the interior control points, $\mathbf{F}_{\mathbf{1}}$ and $\mathbf{F}_{\mathbf{2}}$.

$$\mathbf{F}_{\mathbf{1}} = \frac{1}{3}\{(k_0+k_1)\hat{\mathbf{C}}_{\mathbf{0}} + k_0\hat{\mathbf{C}}_{\mathbf{1}} + 2h_0\vec{\mathbf{H}}_{\mathbf{1}} + h_1\vec{\mathbf{H}}_{\mathbf{0}}\} + \mathbf{H}_{\mathbf{1}}$$

$$\mathbf{F_2} = \frac{1}{3}\{k_1\hat{\mathbf{C}}_\mathbf{0} + (k_0 + k_1)\hat{\mathbf{C}}_\mathbf{1} + h_0\vec{\mathbf{H}}_\mathbf{2} + 2h_1\vec{\mathbf{H}}_\mathbf{1}\} + \mathbf{H}_2$$

The choice of a linear function $k(t)$ was arbitrary. Equation 2.3 is at most a cubic function so the product $k(t)\vec{\mathbf{C}}(t)$ can be at most cubic. $k(t)$ could be chosen to be quadratic, giving a scalar degree of freedom, or $\vec{\mathbf{C}}(t)$ could be chosen to be a quadratic giving a vector degree of freedom. The choice of a quadratic $k(t)$ giving a scalar degree of freedom was used by Jensen [13].

Chiyokura-Kimura constructs a tangent plane field along the boundary by blending cross boundary tangent vectors at each of the triangle vertices to get a cross boundary tangent vector field. The cubic Bézier boundary curves used as input to the scheme plus the constructed tangent vector field completely define the boundary tangent plane field. The interior points that control G^1 cross boundary continuity are then determined by examining the radial derivatives.

2.2.3 Vertex Consistency

Constructing two patches to meet with G^1 continuity is straight-forward. The patches must share the same boundary control points and the interior control points can be set using Farin's [7] conditions or Chiyokura-Kimura's [1] construction.

A more complex problem arises when constructing a network of patches. A patch may require three G^1 joins with its neighbours but these cannot be constructed independently. Continuity along one border may impose a constraint on an interior control point while continuity along a different border may impose a different constraint on the same interior control point. Figure 2.3 shows a cycle of patches centered around one point, illustrating the system of constraints for a simple network. The constraints arising from an even numbered cycle of patches sharing one vertex cannot in general be solved. With odd numbered cycles of patches, every vertex in a mesh will have its own cycle of constraints and these various cycles will be interdependent, requiring a global solution. This is known as the vertex consistency problem.

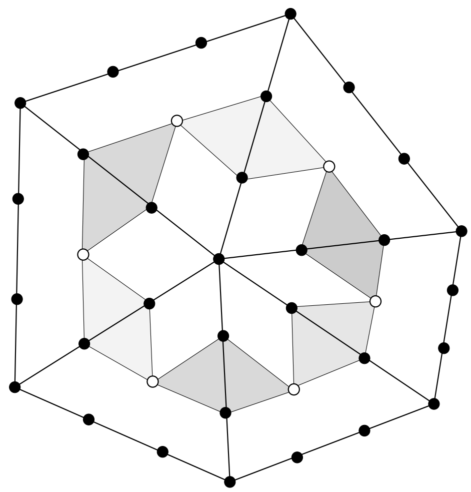

Figure 2.3: The vertex consistency problem with a network of cubic patches. Each pair of shaded panels represents a constraint relating two interior control points. The system of constraints must be satisfied to produce a G^1 surface.

Vertex consistency needs to be solved for two reasons

1. For even numbered cycles of patches, the constraints have no solution in general so a G^1 network of patches cannot be constructed.

2. For odd numbered cycles of patches, the constraints have a solution but a global solution is needed to construct G^1 joins for the entire network of patches. This calculation is computationally expensive and prevents interactive manipulation of the surface.

Rather than solve it, there are two common techniques for bypassing the vertex consistency problem – split domains and rational blends.

Split Domains

The split domain solution to the vertex consistency problem replaces each patch with three patches. Figure 2.4 illustrates how the domain of the original triangle, the macro-triangle, is split into three mini-triangles.

The control points for the mini-triangles are constructed starting from the outside and progressing towards the center. The mini-triangles will have the same boundary control points

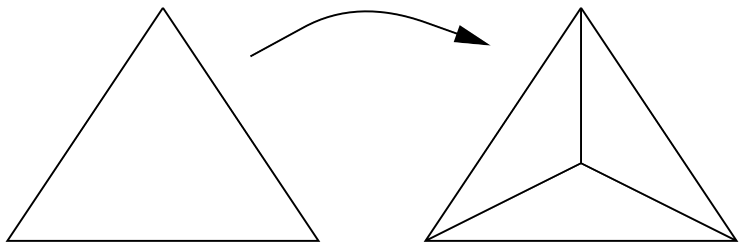

Figure 2.4: Split domain solution for vertex consistency problem. The macro-triangle on the left is split into three mini-triangles on the right.

as the macro-triangle along the outer boundary. The interior points of the mini-triangles are then chosen to give G^1 joins along the outer boundaries. The inner boundaries are then set to give G^1 joins across mini-triangles boundaries.

Splitting the domain helps avoid the vertex consistency problem by moving the problem from the data points to the split point, which will always have an odd numbered cycle of patches – there are three mini-triangles in a macro-triangle. The freedom to set the interior control point of the macro-triangle eliminates the unsolvability of the even numbered cycles of constraints around the macro-triangle corner points and eliminates the need for a global solution for the odd numbered cycles of constraints around the macro-triangle interior point.

Rational Blends

The other standard approach to solving the vertex consistency problem is the rational blend. Multiple triangular Bézier patches are created and each one is designed to handle one aspect of the construction. For example, three patches could be constructed and each patch is constructed to give a G^1 join along only one of the three edges. Figure 2.5 presents a domain space view of the three patches. To evaluate the rational blend interpolant at some parameter value, each of the multiple Bézier patches are evaluated at that parameter value, then an affine combination of those points is taken. The coefficients of the affine combination are rational functions of the parameters hence the name rational blend. Each boundary of the resulting interpolant then has the tangent plane field of one of the multiple patches and therefore has G^1 joins along all the boundaries.

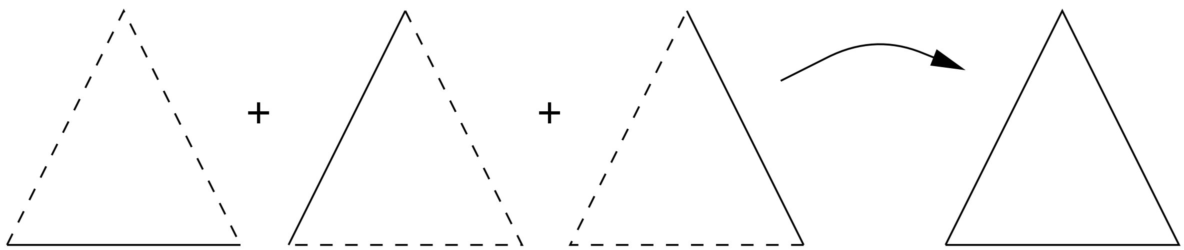

Figure 2.5: Rational blend solution for vertex consistency problem.

The only points on the boundary that have contributions from more than one patch are the corners. The two patches that contribute to tangent plane continuity at a corner will in general have different mixed partial derivatives. Vertex consistency is "solved" by allowing inconsistent mixed partial derivatives at the corner points.

2.2.4 Some Data Fitting Schemes

The following are short descriptions of some scattered data fitting schemes that produce either triangular Bézier patches or hybrid triangular Bézier patches, triangular interpolants constructed from rational blends of triangular Bézier patches. Some of the schemes require boundary curves as inputs while others include boundary curve construction as part of the scheme.

Shirman-Sequin

Shirman and Sequin [20] introduced a split domain scheme that accepts vertex positions, normals, and boundary curves as inputs and produces quartic Bézier patches.

Chiyokura-Kimura's construction is used with the input cubic boundary curves to set the inner control points close to the macro-triangle boundary. The cubic boundary curves are subsequently degree raised to quartic curves. The cross boundary continuity conditions between mini-triangles give equations relating the remaining control points in terms of scalar parameters. These parameters are then set to give certain geometric properties to the interpolant.

Shirman-Sequin's scheme demonstrates one of the difficulties of moving into the parametric setting – the higher degree patches required by the parametric formulation introduce unused degrees of freedom that must either be set using heuristics or left as shape parameters.

Nielson's Scheme

Nielson [19] devised a triangular interpolation scheme that gives tangent plane continuity using rational blends.

Nielson constructed three patches using a side-vertex curve constructor and blended them together. The side-vertex curve constructor,

$$\mathbf{G}(\mathbf{P}, \mathbf{Q}, \mathbf{N_P}, \mathbf{N_Q}),$$

accepts two points and two normals as inputs and produces an interpolating curve. The boundary curves of the three patches are defined by the side-vertex constructor. To evaluate the interior of one of the patches, a point and normal along a boundary curve are evaluated and the side-vertex constructor is used with these and the opposite triangle vertex and associated normal, giving a point in the interior. The following function weights the three points evaluated on the three underlying patches, $\mathbf{q}_0$ $\mathbf{q}_1$ and $\mathbf{q}_2$, giving a point $\mathbf{p}$ lying on the Nielson patch:

$$\mathbf{p}(t_0, t_1, t_2) = a_0(t_0, t_1, t_2)\mathbf{q}_0 + a_1(t_0, t_1, t_2)\mathbf{q}_1 + a_2(t_0, t_1, t_2)\mathbf{q}_2 \tag{2.4}$$

where

$$a_i(t_0, t_1, t_2) = \frac{t_j t_k}{t_i t_j + t_i t_k + t_j t_k}, i \neq j \neq k. \tag{2.5}$$

Gregory's Scheme

Triangular Gregory patches [14] take cubic boundaries as input and use rational blends of interior points to produce an interpolant in a manner similar to the Foley-Opitz method.

Figure 2.6 shows the boundary control points plus all the interior points used in the rational blends. Notice that the patch boundary curves are cubic, which is necessary when using Chiyokura-Kimura's tangent plane field construction. The cubic boundaries are later

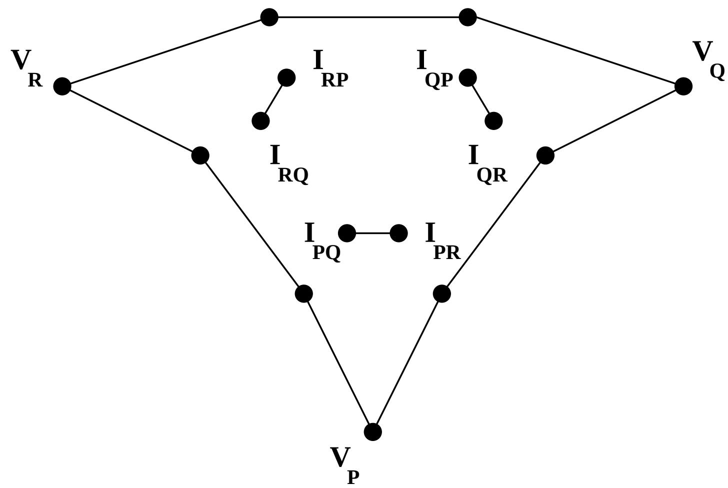

Figure 2.6: Triangular Gregory patch control points.

degree raised to quartic curves once the interior control points have been determined. The interior points are chosen to give a specific tangent plane field along each boundary.

Let b_p, b_q, and b_r be the domain space parameters that weight the vertices $\mathbf{V_p}$,$\mathbf{V_q}$, and $\mathbf{V_r}$ respectively. Then points $\mathbf{I_{ij}}$ and $\mathbf{I_{ik}}$ are the points constructed to ensure cross boundary continuity along the boundary corresponding to $b_i = 0$. To evaluate the patch at (b_p, b_q, b_r), three interior quartic control points would be computed from the six interior control points found in the diagram, using the following formula:

$$\mathbf{I_i} = \frac{b_k(1-b_j)\mathbf{I_{ik}} + b_j(1-b_k)\mathbf{I_ij}}{b_k(1-b_j) + b_j(1-b_k)}$$

Then the quartic patch consisting of these three control points plus the input boundary curves is evaluated at (b_p, b_q, b_r). Thus, along the boundary $b_i = 0$, the interpolant has the same tangent plane field as a quartic Bézier patch with interior control points $\mathbf{I_{ji}}$ and $\mathbf{I_{ki}}$. As with all rational blend schemes the interior of the patch is C^1 but does not in general correspond to any quartic patch.

As an alternative method of evaluation, instead of constructing three quartic patches for every point of evaluation, the blending functions used to construct the $\mathbf{I_i}$'s can be placed over a common denominator allowing the interpolant to be represented as a convex combination of seven different quartic Bézier patches. The blending functions then become degree six polynomials.

2.3 Restricted Data Fitting – Functional Data

In general, the data I am considering can be treated as though sampled from a parametric function of two variables

$$\mathbf{G}(s,t) = \begin{bmatrix} x(s,t) \\ y(s,t) \\ z(s,t) \end{bmatrix}.$$

An interpolation scheme then produces the interpolant $\mathbf{F}(s,t)$ which intersects $\mathbf{G}(s,t)$ at the data points and has the same tangent plane at those points. Both the data and the interpolant are referred to as parametric.

If $\mathbf{G}$ can be restricted to a function parameterized by the plane

$$\mathbf{G}(s,t) = \begin{bmatrix} s \\ t \\ z(s,t) \end{bmatrix},$$

then the interpolant $\mathbf{F}(s,t)$ would have the same form and therefore would be parameterized by the plane. In such case, the data and the interpolant are referred to as *functional.*

An important property of functional patches that arises from the natural parameterization is that the x and y coordinates of the control points are completely determined by the x and y coordinates of the triangle vertices – the control points of a patch must be evenly distributed over the domain triangle. This means that when constructing cubic patches, the triangle vertices and normals completely determine the boundary curves.

In the parametric setting, the x, y, and z coordinates of the control points are free variables. In the functional setting, only the z coordinates of the control points are free variables.

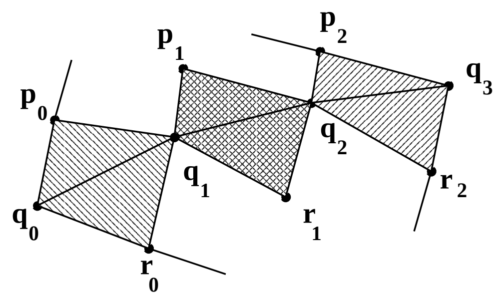

Figure 2.7: C^1 continuity conditions

2.3.1 C^1 Continuity

Functional interpolation schemes typically produce C^1 surfaces. A simple geometric condition can be given that ensures C^1 continuity for the boundary separating two patches, shown in Figure 2.7 for the case of cubics. q_0, q_1, q_2, and q_3 are the common boundary control points. p_0, p_1, p_2 and r_0, r_1, r_2 are the two "parallel" rows of control points next to the common boundary.

Each pair of marked triangles must be coplanar for C^1 continuity. In other words, p_0, q_0, r_0, and q_1 must be coplanar, p_1, q_1, r_1, and q_2 must be coplanar, and p_2, q_2, r_2, and, q_3 must be coplanar. The condition is similar for non-cubic patches. Each pair of triangles along the shared boundary must be coplanar.

2.3.2 C^2 Continuity

Figure 2.8 illustrates the necessary conditions for C^2 continuity between two cubic functional Bézier patches.

Suppose that the barycentric coordinates of $\mathbf{W_2}$ with respect to $\mathbf{V_0}$, $\mathbf{V_1}$, and $\mathbf{V_2}$ are r, s, and t respectively, i.e., $\mathbf{W_2} = r\mathbf{V_0} + s\mathbf{V_1} + t\mathbf{V_2}$. Also, suppose that the barycentric coordinates of $\mathbf{V_2}$ with respect to $\mathbf{W_0}$, $\mathbf{W_1}$, and $\mathbf{W_2}$ are $\overline{r}$, $\overline{s}$, and $\overline{t}$ respectively, i.e., $\mathbf{V_2} = \overline{r}\mathbf{W_0} + \overline{s}\mathbf{W_1} + \overline{t}\mathbf{W_2}$.

For the two patches to meet with C^2 continuity the following equations must hold

$$r\mathbf{q_0} + s\mathbf{q_1} + t\mathbf{p_0} = \overline{r}\mathbf{r_0} + \overline{s}\mathbf{r_1} + \overline{t}\mathbf{s_0} \tag{2.6}$$

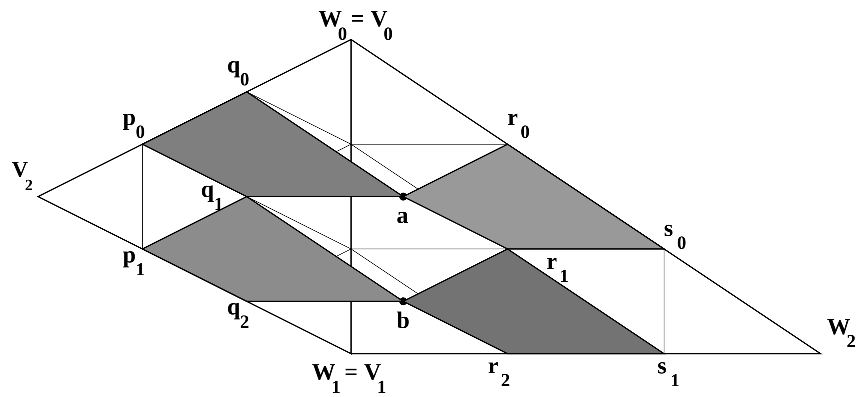

Figure 2.8: C^2 continuity conditions

$$r\mathbf{q_1} + s\mathbf{q_2} + t\mathbf{p_1} = \overline{r}\mathbf{r_1} + \overline{s}\mathbf{r_2} + \overline{t}\mathbf{s_1} \tag{2.7}$$

In other words, the planes hi-lighted by the shaded regions in the figure must have the same z-value at the points marked **a** and **b**.

2.3.3 Clough-Tocher

The Clough-Tocher [2] [12] interpolation scheme is a classic solution to the functional data fitting problem. It has been around for several decades and was originally used as a technique in finite element analysis. Clough-Tocher takes points and corresponding normals and produces a split domain cubic interpolant with quadratic precision, i.e., given data sampled from a quadratic function, the scheme will reproduce the original quadratic function.

The domain macro-triangle is split into three mini-triangles, shown in Figure 2.9, all of which are the domains of cubic patches.

Figure 2.10 shows the order in which control points are constructed. The points marked 0 are inputs and have normals associated with each of them. Points marked 1 are determined by the vertex positions and vertex normals using Hermite interpolation. Points marked 2 are then calculated to ensure C^1 continuity across the macro-triangle boundary. The points marked 3 are determined by the C^1 continuity conditions across mini-triangle boundaries.

The only degrees of freedom in the entire scheme, the settings of the points marked 2, are set to give linearly varying cross boundary derivatives across the macro-triangle boundaries.

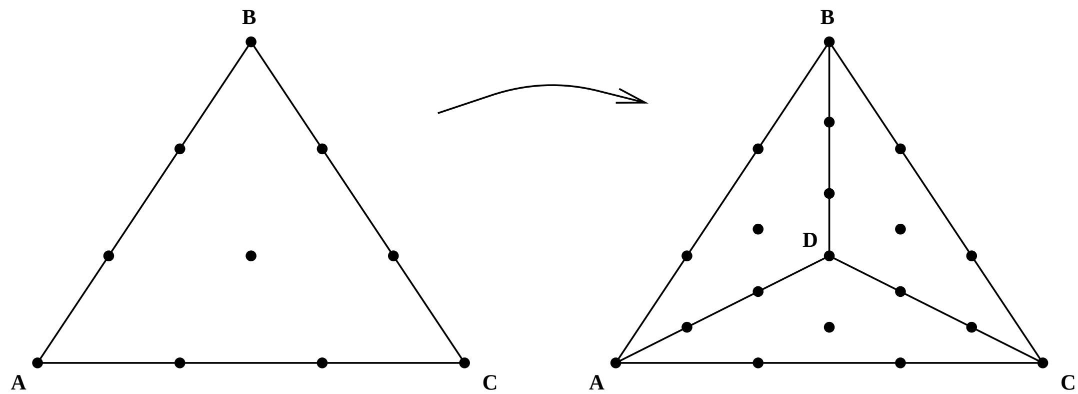

Figure 2.9: Top-down view of triangle subdivision

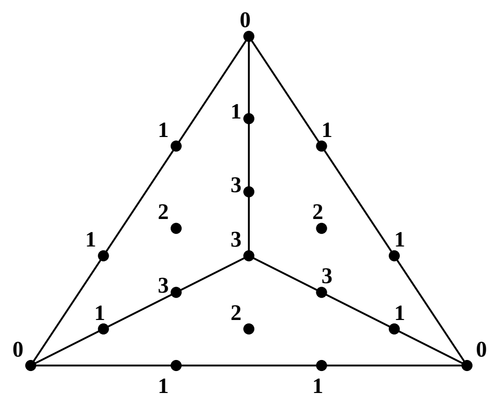

Figure 2.10: Order of calculation of Clough-Tocher control points

The choice of linearly varying derivatives results in a C^1 cubic interpolant with quadratic precision.

Chapter 3

Boundary Curves and Cross Boundary Continuity

My goal is to improve the surface quality of existing parametric interpolants. There are two components to patch construction that can be examined separately – boundary curve construction and cross boundary continuity construction.

Functional cubic boundary curve construction is completely determined by corner positions and normals. In contrast, parametric boundary curve construction is more problematic. Mann [16] has concluded that poor boundary curve construction is the primary reason for poor visual quality of parametric triangular interpolants.

My approach to improving parametric surface quality will be to study interpolation schemes in the functional setting and to transfer a construction to the parametric setting. This means that boundary curves will not be an initial consideration since they are completely determined in the functional setting. Instead, I will concentrate on a cross boundary construction and investigate whether it can be transferred to the parametric setting.

I will consider a functional interpolation scheme, the Foley-Opitz [10] scheme, that improves on Clough-Tocher. The key component of the scheme is the cross boundary continuity construction – not only are patch joins C^1 but the patch has cubic precision. There is visual improvement over the C^1 surfaces with quadratic precision produced by Clough-Tocher.

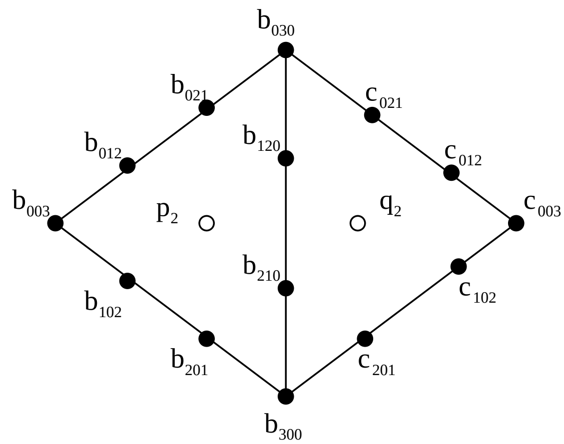

Figure 3.1: Domain control net for the Foley-Opitz hybrid Bézier patch.

3.1 The Foley-Opitz Scheme

Foley and Opitz [10] present a method for interpolation of scattered data above the plane using a "hybrid" cubic Bézier patch that is based on Nielson's scheme. The cubic patch boundary is determined by the triangle vertices and normals. The calculation of the three inner control points using a new C^1 cross boundary construction is introduced that gives cubic precision, i.e., when given data sampled from a cubic surface the scheme reproduces that cubic surface.

Figure 3.1 shows the domain control net for two neighbouring triangles. p_2 is one of the three interior control points associated with the left triangle and q_2 is one of the three interior control points associated with the right triangle. The following is the Foley-Opitz method for computing p_2.

Let r,s, and t be the barycentric coordinates of c_{003} with respect to b_{003}, b_{030}, and b_{003}. If both patches form a single cubic, then from subdividing Bézier cubics it can be shown that

$$c_{102} = r^2 b_{300} + 2rsb_{210} + 2rtb_{201} + s^2 b_{120} + 2stp_2 + t^2 b_{102} \tag{3.1}$$

$$c_{012} = r^2 b_{210} + 2rsb_{120} + 2stb_{021} + s^2 b_{030} + 2rtp_2 + t^2 b_{012} \tag{3.2}$$

Similarly, if u, v, and w are the barycentric coordinates of b_{003} with respect to c_{003}, c_{030}, and

c_{003} then

$$b_{102} = u^2 c_{300} + 2uvc_{210} + 2uwc_{201} + v^2 c_{120} + 2vwq_2 + w^2 c_{102} \tag{3.3}$$

$$b_{012} = u^2 c_{210} + 2uvc_{120} + 2vwc_{021} + v^2 c_{030} + 2uwq_2 + w^2 c_{012} \tag{3.4}$$

Equations 3.1 and 3.2 are two of the three equations obtained during the second last step of the deCasteljau algorithm when evaluating the patch given by b_{300}, b_{030}, and b_{003} at (r, s, t), the barycentric coordinates of c_{003} with respect to $(b_{300}, b_{030}, b_{003})$.

Foley and Opitz show that the over-determined system of equations for p_2 in Equations 3.1 and 3.2 always has a solution and similarly that q_2 can always be determined from Equations 3.3 and 3.4.

p_2 and q_2 as calculated above ensure that the two triangles have a C^1 join along their common border. Identical calculations would be made to ensure C^1 continuity across the remaining two edges giving three settings for the interior control points of each of the two patches.

Nielson's rational blend function, Equation 2.5, is used to weight the three calculated interior control points, giving

$$\mathbf{b}_{111}(t_0, t_1, t_2) = a_0(t_0, t_1, t_2)\mathbf{p}_0 + a_1(t_0, t_1, t_2)\mathbf{p}_1 + a_2(t_0, t_1, t_2)\mathbf{p}_2 = \mathbf{p}, \tag{3.5}$$

a point that is then treated as the single interior control point of a cubic patch. This rational blend introduces removable singularities and inconsistent mixed partial derivatives at the triangle vertices.

To see why the patch has cubic precision, consider p_0, p_1, and p_2, the interior points corresponding to the edges $b_{300}b_{030}$, $b_{300}b_{003}$, and $b_{030}b_{003}$ respectively. Suppose that the three points, b_{300}, b_{030}, and b_{003}, and corresponding normals are taken from a cubic surface. The three calculated interior points will be identical since the cross-boundary construction assumed that the data was sampled from a cubic surface. The resulting blend of the three interior control points will give the interior control point for a cubic surface, meaning that the Foley-Opitz patch becomes a cubic Bézier patch. Thus, the Foley-Opitz method has cubic precision.

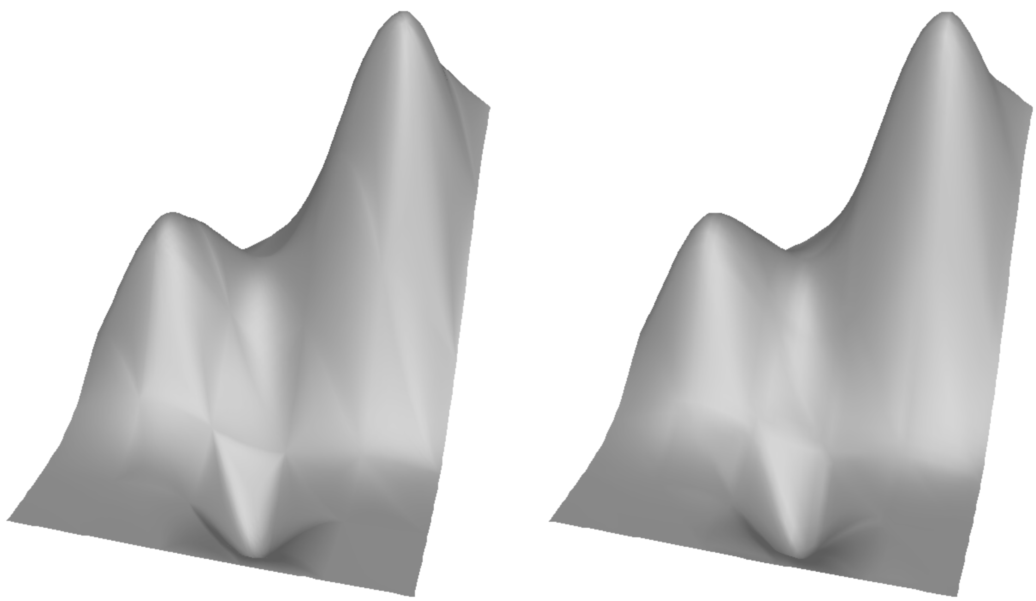

Figure 3.2: The Franke data set interpolated with Clough-Tocher patches, left, and Foley-Opitz patches, right.

Figure 3.2 shows the Franke data set [11] interpolated with Clough-Tocher and Foley-Opitz patches. The smoothing out of patch boundaries is the significant improvement in the surface quality. The underlying triangular patches are visible throughout the Clough-Tocher interpolant and are noticeable as creases surrounding triangular regions of the surface. The Foley-Opitz interpolant is generally smoother but still has some visible patch boundary creases.

3.2 Combining Foley-Opitz with Clough-Tocher

Both Foley-Opitz and Clough-Tocher provide solutions to the data interpolation problem. However, each scheme has both strengths and weaknesses. The Foley-Opitz scheme gives a C^1 interpolant with cubic precision at the cost of using a rational blend, a degree five over degree two rational polynomial. The rational blend introduces removable singularities and inconsistent mixed partial derivatives at the triangle vertices. Using the split domain technique, Clough-Tocher produces a piecewise cubic patch with quadratic precision.

Mann [17] has combined these two schemes. In calculating the midpoints of the three micro-triangles in Clough-Tocher the choice of conditions ensuring quadratic precision was arbitrary. Simply exchange Clough-Tocher's cross-boundary conditions for those of Foley-Opitz. The added complication over and above the standard calculation required for Clough-Tocher will be the inspection of neighbouring triangles. Clough-Tocher uses a blend of the boundary control points to set the midpoint while Foley-Opitz uses information from the neighbouring triangle in addition to the boundary control points.

The resulting patch will be piecewise cubic and will have cubic precision. Given data sampled from a cubic surface, the original surface will be reproduced using this method. The patch will not have the inconsistent mixed partial derivatives found with the Foley-Opitz scheme or Nielson's scheme.

3.3 Other Cross Boundary Schemes

There are other functional cross boundary schemes for cubics [17] which I will briefly describe.

Farin [6] devised a split domain cubic C^1 scheme that minimizes C^2 discontinuity between mini-triangles across macro-triangle boundaries. A macro-triangle is constructed to give quadratic precision then is subdivided into three mini-triangles. The interior control points of the mini-triangles are adjusted to minimize C^2 discontinuity across the macro-triangle boundaries. The remaining mini-triangle boundary control points are then set as in Clough-Tocher. This scheme has quadratic precision.

Mann [17] has modified Farin's scheme to give cubic precision by constructing three macro-triangles, each of which minimizes C^2 discontinuity across one boundary. One mini-triangle is retained from each patch after subdividing and leftover mini-triangle boundary control points are set as in Clough-Tocher. A Clough-Tocher scheme using the Foley-Opitz technique and this modified Farin method are nearly identical. Both methods have cubic precision, but while in general both yield better surfaces than the original Farin technique, there are times when the original Farin technique produces surfaces of better quality.

Since the difference between these schemes and the Foley-Opitz is minor, I did not test any of them in my surface construction technique. I did not consider any iterative cross boundary techniques [6] [8].

Chapter 4

A New Interpolation Scheme

The problem with local data fitting in the parametric setting is that existing schemes produce visually unsatisfying interpolants. Mann [16] has published a survey of parametric scattered data fitting schemes using triangular interpolants. The conclusion reached in that survey is that the methods available at the time produced interpolants of roughly the same quality, and that quality was poor, despite all the interpolants being G^1. Even though different surface construction methods were used – split domain and rational blend – the results were uniformly mediocre. This was attributed to the one step of surface construction common to all the schemes – boundary construction. Many schemes require boundary curves as input to the interpolation scheme while others provide some method of construction. The construction of boundary Bézier curves does not exploit the structure of the patch or naturally complement the patch.

One of the underlying problems behind the poor surface quality and the boundary curve construction is the number of degrees of freedom. Unlike curve construction, in which the degrees of freedom roughly equal the number of conditions imposed, the surface construction problem does not give as efficient a solution. In this setting, parametric cubic patches cannot be joined with G^1 continuity, and the need for higher degree patches results in many excess degrees of freedom, which are typically set using heuristics or are left as shape parameters for a modeler. The ideal interpolation scheme would use up all the excess degrees of freedom to satisfy geometric constraints and, in doing so, produce a fair surface.

In light of the problems with parametric patches, I have decided to approach the problem of data interpolation from the better understood domain of functional schemes and have tried to "parameterize" one of these schemes.

4.1 Overview of the New Scheme

My goal is to create a parametric version of Foley-Opitz so as to bring its good surface quality to the parametric setting. Converting Clough-Tocher or a related split domain scheme to the parametric setting would require degree raising to quartic patches or higher at best. Converting a rational blend functional scheme to the parametric setting is easier, promises a greater degree of success, and preserves the structure of a cubic patch.

The Foley-Opitz cross boundary construction relies on a natural parameterization between patch pairs since the barycentric coordinates of neighbouring patches with respect to each other are key in determining the tangent plane fields. Barycentric coordinates are always taken with respect to a plane that parameterizes the patches. Therefore, some association between neighbouring patch domains must be made in order to use the Foley-Opitz tangent plane field construction.

My approach is to choose a plane for each patch pair, project the control points of both patches onto the plane, and then perform the Foley-Opitz C^1 construction. Three sets of control points are calculated for each patch – each set representing a C^1 construction along one triangle edge. The three sets of control points will share the same triangle vertices but in general differ in the rest of the boundary and interior control points. Unfortunately, the rational blend used by Foley-Opitz in the functional setting only allows for a blend of the interior control points and not the boundary control points.

However, having multiple sets of boundaries can be overcome by treating the boundary points in the same manner as the interior points – both boundary and interior control points are blended to produce the final interpolant. Figure 4.1 shows the domain control net for a parametric version of the Foley-Opitz scheme. The structure is similar to Foley-Opitz – the control points are organized as the control points of a cubic triangular Bézier patch except

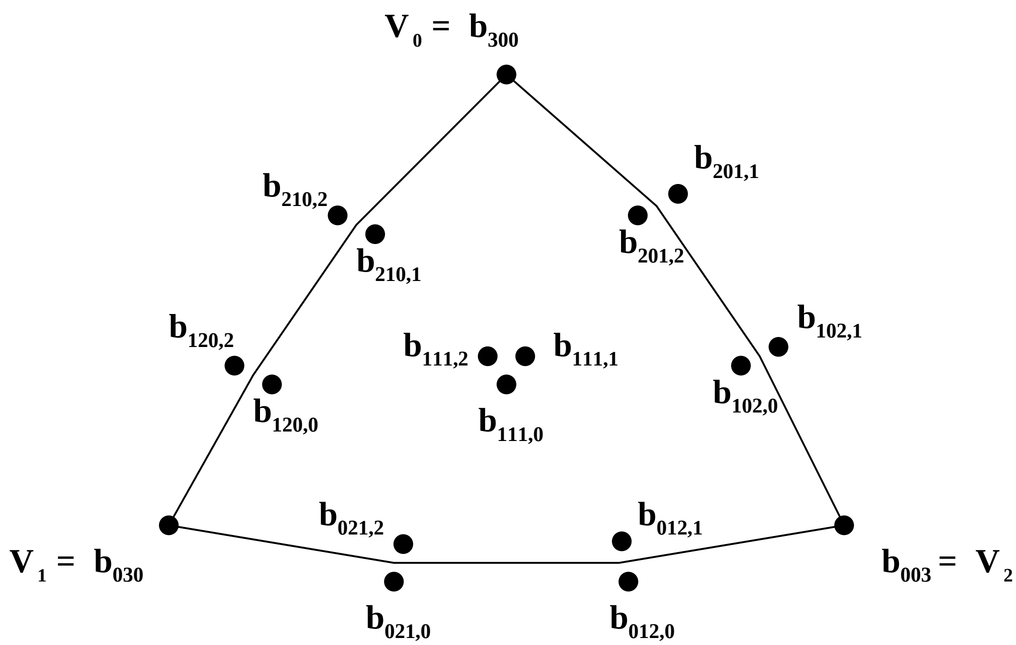

Figure 4.1: Domain control net for a parametric Foley-Opitz patch.

that a group of parametric Foley-Opitz control points correspond to a single regular cubic control point.

The control points are taken from three different Bézier patches and characterize the three boundary tangent plane fields. Each of the three patches share the same triangle vertices, $\mathbf{b_{300}}, \mathbf{b_{030}}$, and $\mathbf{b_{003}}$, but in general do not share any other control points. By contributing a set of control points each patch determines the tangent plane field along one edge. Figures 4.2, 4.3, and 4.4 show the contributions from each of three Bézier patches to the parametric Foley-Opitz patch. The control points in the figures determine the tangent plane field along the $\mathbf{b_{030}b_{003}}$, $\mathbf{b_{300}b_{003}}$, and $\mathbf{b_{300}b_{030}}$ edges respectively.

In the construction of the tangent plane field along a particular boundary only two parametric Foley-Opitz patches are involved and consequently only two Bézier patches. Each Bézier patch contributes seven control points to a parametric Foley-Opitz patch, as in Figure 4.2. These control points determine the tangent plane field along that boundary. To construct the seven points, some plane is chosen as a natural parameterization for the two Bézier patches allowing the construction as in the Foley-Opitz functional scheme. Once the

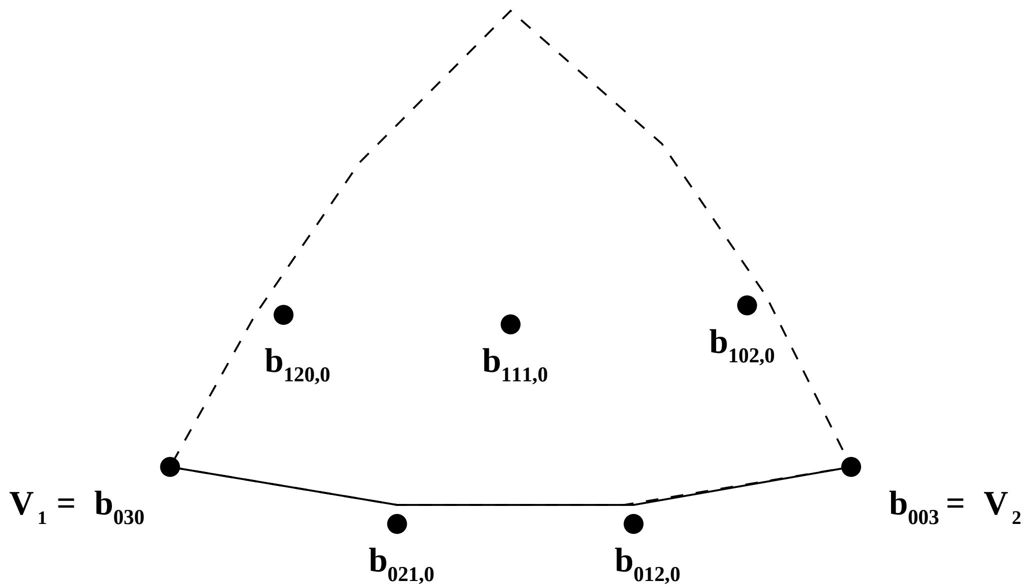

Figure 4.2: Tangent plane field control points along the $\mathbf{b}_{030}\mathbf{b}_{003}$ edge.

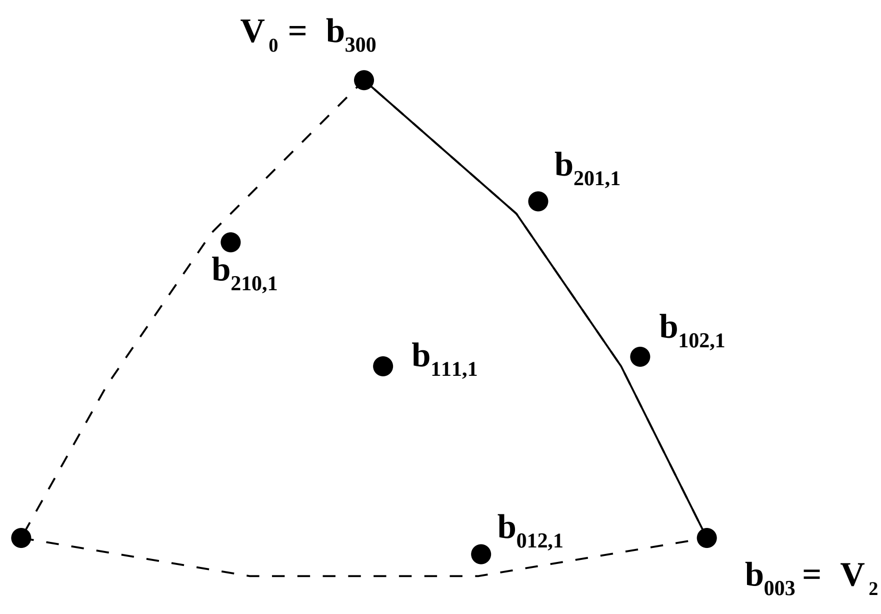

Figure 4.3: Tangent plane field control points along the $\mathbf{b}_{300}\mathbf{b}_{003}$ edge.

plane is chosen, then the Bézier patch control points are completely determined by the triangle vertices and the associated normals. Hermite interpolation over a plane completely determines the cubic boundary curves and the Foley-Opitz cross boundary construction de-

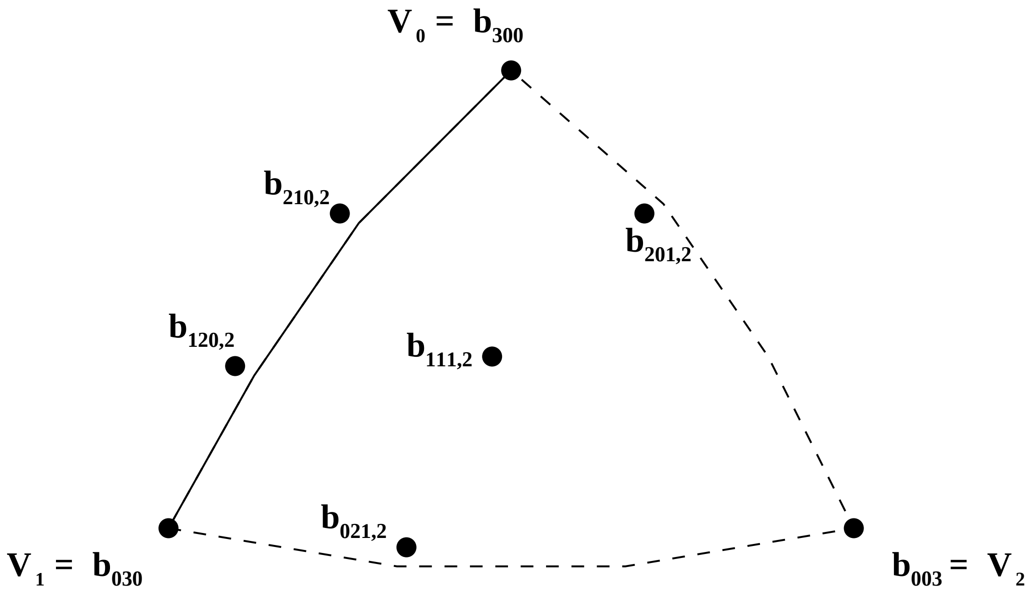

Figure 4.4: Tangent plane field control points along the $\mathbf{b_{300}b_{030}}$ edge.

termines the interior control points.

There are some degrees of freedom in the choice of a plane and also some restrictions. The orientation, not the position, of the plane determines the positions of the control points thus giving two rotational degrees of freedom. However, treating the plane as a function that maps a set of points and normals to two angles is awkward. The plane should be constructed geometrically from the given information – triangle vertices and normals. The two Bézier patches must not have overlapping domains on the plane so the orientation is restricted to being "underneath" both patches.

One failsafe method of choosing a feasible plane is to take the plane that is perpendicular to the bisecting plane of the two neighbouring triangles and that also contains their common edge. Figure 4.5 shows the bisecting plane for the two triangles $\mathbf{PQR}$ and $\mathbf{SQR}$. The triangle is bisected in the sense that $\angle\mathbf{PRT} = \angle\mathbf{SRT}$ and $\angle\mathbf{PQT} = \angle\mathbf{SQT}$ for any point $\mathbf{T}$ on the bisecting plane. The final plane, Figure 4.6, contains the edge $\mathbf{QR}$ and is perpendicular to the bisecting plane. Note that $\overline{\mathbf{P}}$, the projection of the triangle vertex $\mathbf{P}$, will never lie in the the triangle $\overline{\mathbf{S}}\mathbf{QR}$ so the patch domains will never overlap.

Another possibility is to use the information provided by the normals at the triangle

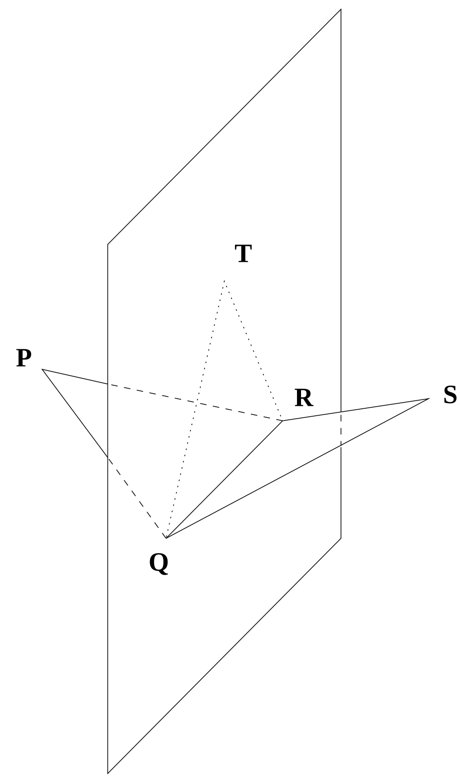

Figure 4.5: The bisecting plane for triangles **PQR** and **SQR**. Angles **PRT** = **SRT** and angles **PQT** = **SQT** for any point **T** on the bisecting plane.

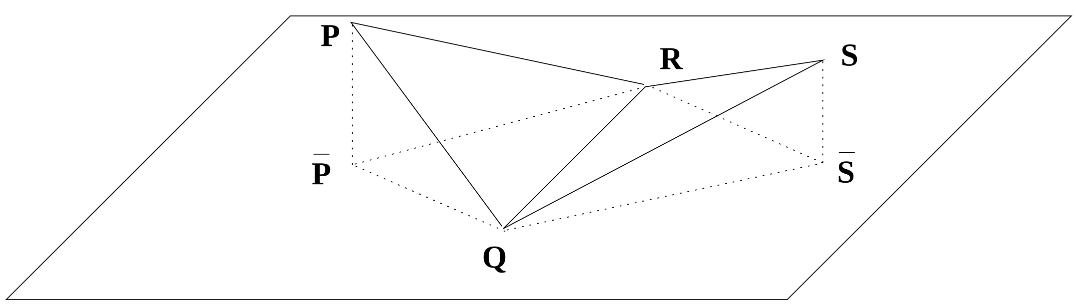

Figure 4.6: Plane used to parameterize neighbouring Bézier patch pairs.

vertices to construct the plane. The normals give information about the shape of the function from which the data was sampled making them ideal as input for the construction of the parameterization plane.

Once the control points from three Bézier patches are calculated they can then be used

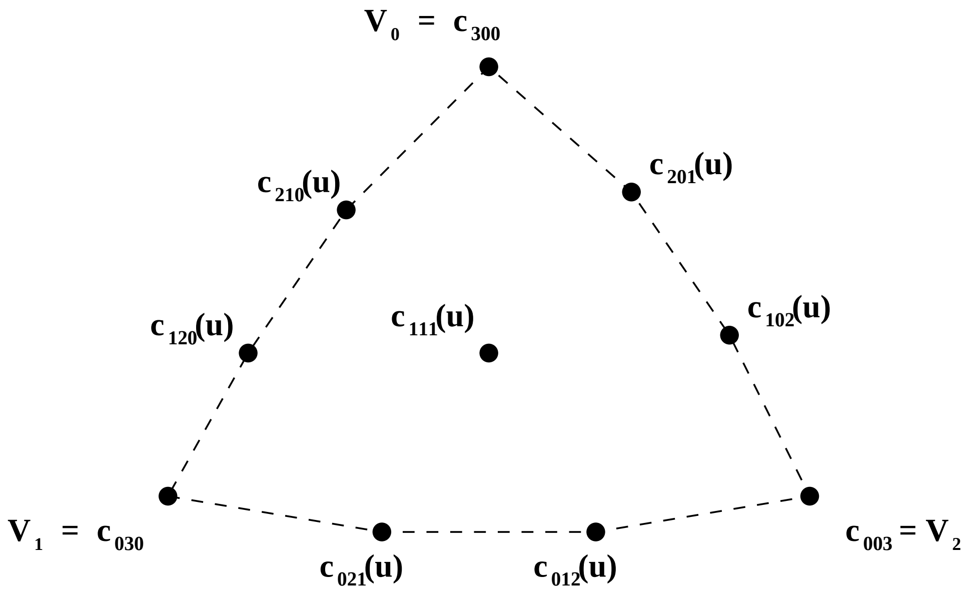

Figure 4.7: Control net after blending parametric Foley-Opitz control points.

to formulate the parametric Foley-Opitz patch. However, blending is not as straightforward as in the functional case – boundary points are included in the blend. Figure 4.7 illustrates the control net for the parametric patch. Each point $\mathbf{c_{ijk}}$ is a rational blend of the associated control points from the three Bézier patches shown in Figure 4.1. The points $\mathbf{c_{300}}, \mathbf{c_{030}}$, and $\mathbf{c_{003}}$ are constants, the remaining boundary control points are rational blends of two Bézier patch boundary control points and three Bézier patch interior control points. The coefficients used to blend the $\mathbf{b_{ijk}}$ control points are functions of the patch parameters so that different $\mathbf{c_{ijk}}$ will be used at each point of the evaluation.

When evaluating along the edges, the blend formulation must preserve the important properties of the three underlying Bézier patches, boundary curves and cross boundary derivatives. The corner control points are constants so no blending function is needed. The blending function for the interior control points is the same as used in the Foley-Opitz functional construction.

$$\mathbf{c_{111}}(\mathbf{u}) = w_0(\mathbf{u})\mathbf{b_{111,0}} + w_1(\mathbf{u})\mathbf{b_{111,1}} + w_2(\mathbf{u})\mathbf{b_{111,2}}$$

where

$$w_i(u_0, u_1, u_2) = \frac{u_j u_k}{u_0 u_1 + u_0 u_2 + u_1 u_2}, \ i \neq j \neq k$$

The boundary control points are blended to give two properties:

1. When evaluated along a boundary the parametric Foley-Opitz control points become the control points of one of the three Bézier patches

2. The tangent plane field of the parametric Foley-Opitz patch along a boundary matches the tangent plane field along the same boundary of one of the three Bézier patches.

An asymmetric blend of the following form has both of these properties

$$h_{ij}(u_0, u_1, u_2) = \frac{(1 - u_i){u_j}^2}{(1 - u_i){u_j}^2 + (1 - u_j){u_i}^2} \tag{4.1}$$

This blending function is used to weight all the non-vertex boundary Bézier control points. The blends for the $\mathbf{c_{ijk}}$ control points have a similar structure but differ in the exact blend functions used. For example, the $\mathbf{V_1 V_2}$ edge control points would be

$$\mathbf{c_{012}}(\mathbf{u_0}, \mathbf{u_1}, \mathbf{u_2}) = h_{01}(u_0, u_1, u_2)\mathbf{b_{012,0}} + h_{10}(u_0, u_1, u_2)\mathbf{b_{012,1}}$$

$$\mathbf{c_{021}}(\mathbf{u_0}, \mathbf{u_1}, \mathbf{u_2}) = h_{02}(u_0, u_1, u_2)\mathbf{b_{021,0}} + h_{20}(u_0, u_1, u_2)\mathbf{b_{021,2}}$$

Evaluation along the $\mathbf{V_1 V_2}$ edge $u_0 = 0$ gives

$$\mathbf{c_{012}}(\mathbf{0}, \mathbf{u_1}, \mathbf{u_2}) = h_{01}(0, u_1, u_2)\mathbf{b_{012,0}} + h_{10}(0, u_1, u_2)\mathbf{b_{012,1}} = \mathbf{b_{012,0}}$$

$$\mathbf{c_{021}}(\mathbf{0}, \mathbf{u_1}, \mathbf{u_2}) = h_{02}(0, u_1, u_2)\mathbf{b_{021,0}} + h_{20}(0, u_1, u_2)\mathbf{b_{021,2}} = \mathbf{b_{021,0}}$$

since Equation 4.1 gives $h_{0j}(0, u_1, u_2) = 1$ and $h_{i0}(0, u_1, u_2) = 0$. The $\mathbf{V_1 V_2}$ boundary curve is the cubic Bézier curve given by the control points $\mathbf{b_{030}}$, $\mathbf{b_{021,0}}$, $\mathbf{b_{012,0}}$, and $\mathbf{b_{003}}$ which is suggested by Figure 4.2.

4.2 Details of the New Scheme

Control Points

I will introduce a notation for parametric Foley-Opitz patches that allows for a concise definition of the patch. Figure 4.7 shows that the structure of the control net for a parametric Foley-Opitz patch is similar to the case of a cubic triangular Bézier patch and Figure 4.1 shows how the notation is also similar. The control points for a parametric Foley-Opitz patch are

$$\mathbf{b_{300}}, \mathbf{b_{030}}, \mathbf{b_{003}}$$

$$\mathbf{b_{111,0}}, \mathbf{b_{111,1}}, \mathbf{b_{111,2}}$$

$$\mathbf{b}_{\sigma\mathbf{(012)},\sigma_\mathbf{0}}, \mathbf{b}_{\sigma\mathbf{(012)},\sigma_\mathbf{1}}.$$

The corner control points $\mathbf{b_{300}}, \mathbf{b_{030}}, \mathbf{b_{003}}$ are exactly the same as in a cubic Bézier patch. The interior $\mathbf{b_{111,i}}$ points, and the way in which they are subsequently blended, are taken from the paper of Foley and Opitz. The points that are subscripted with $\sigma(012), \sigma_i$ represent a new notation that allows the patch definition to have the same form as a standard cubic Bézier patch.

$\sigma(012)$ represents an arrangement of the numbers 012. σ_0 is the position of 0 in the arrangement, and σ_1 is the position of 1 in the arrangement. For example, choosing $\sigma(012) =$ 210 would give

$$\sigma_0 = 2$$

$$\sigma_1 = 1$$

$$\mathbf{b}_{\sigma\mathbf{(012)},\sigma_\mathbf{0}} = \mathbf{b_{210,2}}$$

$$\mathbf{b}_{\sigma\mathbf{(012)},\sigma_\mathbf{1}} = \mathbf{b_{210,1}}$$

The difference in notation between these points and regular cubic Bézier control points is the addition of the second subscript. The new subscript indicates which parameter, and consequently which edge, is most closely associated with the control point. As will be demonstrated shortly, the σ_0 control points control C^0 continuity, and the σ_1 control points control G^1 continuity.

The control points $\{\mathbf{b_{300}}, \mathbf{b_{030}}, \mathbf{b_{003}}, \mathbf{b_{\vec{i},j}}\}$ are blended together to form the ten control points of a standard cubic Bézier patch, labeled as $\mathbf{c_{\vec{i}}}(\mathbf{u})$. The Bézier patch defined by the $\mathbf{c_{\vec{i}}}$ is then evaluated at $\mathbf{u}$. The concise definition given by the new notation is

$$\mathbf{F}(\mathbf{u}) = \sum_{|\vec{i}|=3} B^3_{\vec{i}}(\mathbf{u})\mathbf{c_{\vec{i}}}(\mathbf{u})$$

There are then three types of $\mathbf{b_{\vec{i}}}$ control points: triangle vertices, boundary control points that are not triangle vertices, and interior control points. The corner control points $\mathbf{b_{300}}$, $\mathbf{b_{030}}$, and $\mathbf{b_{003}}$ are not blended so

$$\mathbf{c_{300}}(\mathbf{u}) = \mathbf{b_{300}},\ \mathbf{c_{030}}(\mathbf{u}) = \mathbf{b_{030}},\ \mathbf{c_{003}}(\mathbf{u}) = \mathbf{b_{003}}.$$

The center control points $\mathbf{b_{111,0}}$, $\mathbf{b_{111,1}}$, and $\mathbf{b_{111,2}}$, are blended as in the Foley-Opitz scheme [10] using the blend functions

$$w_i(u_0, u_1, u_2) = \frac{u_j u_k}{u_0u_1 + u_0u_2 + u_1u_2},\ \ i \neq j \neq k,$$

and the center control point is

$$\mathbf{c_{111}}(\mathbf{u}) = w_0(\mathbf{u})\mathbf{b_{111,0}} + w_1(\mathbf{u})\mathbf{b_{111,1}} + w_2(\mathbf{u})\mathbf{b_{111,2}}.$$

The boundary control points that are not triangle vertices are blended in the following manner

$$\mathbf{c_{\sigma(012)}}(\mathbf{u}) = h_{\sigma_0\sigma_1}(\mathbf{u})\mathbf{b_{\sigma(012),\sigma_0}} + h_{\sigma_1\sigma_0}(\mathbf{u})\mathbf{b_{\sigma(012),\sigma_1}}, \tag{4.2}$$

where

$$h_{ij}(\mathbf{u}) = \frac{(1-u_i){u_j}^2}{(1-u_i){u_j}^2 + (1-u_j){u_i}^2}. \tag{4.3}$$

The boundary control points are thus

$$\mathbf{c_{021}}(\mathbf{u}) = \frac{(1-u_0){u_2}^2\mathbf{b_{021,0}} + (1-u_2){u_0}^2\mathbf{b_{021,2}}}{(1-u_0){u_2}^2 + (1-u_2){u_0}^2}$$

$$\mathbf{c_{012}}(\mathbf{u}) = \frac{(1-u_0){u_1}^2\mathbf{b_{012,0}} + (1-u_1){u_0}^2\mathbf{b_{012,1}}}{(1-u_0){u_1}^2 + (1-u_1){u_0}^2}$$

$$\mathbf{c_{201}}(\mathbf{u}) = \frac{(1-u_1){u_2}^2\mathbf{b_{201,1}} + (1-u_2){u_1}^2\mathbf{b_{201,2}}}{(1-u_1){u_2}^2 + (1-u_2){u_1}^2}$$

$$\mathbf{c_{102}}(\mathbf{u}) = \frac{(1-u_1){u_0}^2\mathbf{b_{102,1}} + (1-u_0){u_1}^2\mathbf{b_{102,0}}}{(1-u_1){u_0}^2 + (1-u_0){u_1}^2}$$

$$\mathbf{c_{210}}(\mathbf{u}) = \frac{(1-u_2){u_1}^2\mathbf{b_{210,2}} + (1-u_1){u_2}^2\mathbf{b_{210,1}}}{(1-u_2){u_1}^2 + (1-u_1){u_2}^2}$$

$$\mathbf{c_{120}}(\mathbf{u}) = \frac{(1-u_2){u_0}^2\mathbf{b_{120,2}} + (1-u_0){u_2}^2\mathbf{b_{120,0}}}{(1-u_2){u_0}^2 + (1-u_0){u_2}^2}$$

Evaluating the patch at the boundary will reveal some information about the structure of the patch and explain some of the notation used to describe the points. Note that when evaluated along the edge $u_{\sigma_0} = 0$, Equation 4.2 becomes

$$\Big[\mathbf{c}_{\sigma(\mathbf{012})}(\mathbf{u})\Big]_{u_{\sigma_0}=0} = \mathbf{b}_{\sigma(\mathbf{012}),\sigma_\mathbf{0}}$$

since Equation 4.3 gives

$$\Big[h_{\sigma_0\sigma_1}(\mathbf{u})\mathbf{b}_{\sigma(\mathbf{012}),\sigma_\mathbf{0}}\Big]_{u_{\sigma_0}=0} = 1, \ \Big[h_{\sigma_1\sigma_0}(\mathbf{u})\mathbf{b}_{\sigma(\mathbf{012}),\sigma_\mathbf{0}}\Big]_{u_{\sigma_0}=0} = 0$$

The triangle vertices plus the control points subscripted with σ_0, $\mathbf{b}_{\sigma(\mathbf{012}),\sigma_\mathbf{0}}$, determine the parametric Foley-Opitz patch boundary. This continuity is completely independent of control points with a σ_1 subscript, $\mathbf{b}_{\sigma(\mathbf{012}),\sigma_\mathbf{1}}$, since Equation 4.2 gives these points zero weight along the patch boundary.

Cross Boundary Derivative and Tangent Plane Field

The tangent plane field is determined by the boundary curve derivative and the cross boundary derivative. The boundary curve derivatives are the derivative of the three Bézier curves that form the borders of the patch. The cross boundary derivatives of the patch are directional derivatives with respect to a domain vector, $\mathbf{d} = (d, e, f)$, and evaluated at a point on one of the three patch boundaries, $u_0 = 0$, $u_1 = 0$, or $u_2 = 0$. $\mathbf{d}$ must not be parallel to the boundary containing the point of evaluation unless the point is one of the three triangle vertices, $\mathbf{b_{300}}$, $\mathbf{b_{030}}$, or $\mathbf{b_{003}}$. This section calculates the cross boundary derivative of the patch $\mathbf{F}(u_0, u_1, u_2)$ and draws conclusions about the structure of the patch.

The patch formulation is symmetric in form with respect to the parameters u_0, u_1, and u_2 so that the cross boundary derivative need only be computed for one of the $\mathbf{b_{300}}\mathbf{b_{030}}$,

$\mathbf{b_{300}b_{003}}$, or $\mathbf{b_{030}b_{003}}$ edges. The derivative will be calculated on the $\mathbf{b_{300}b_{030}}$ edge at the point $\mathbf{u} = (t, 1-t, 0), t \in (0,1)$. The general directional derivative vector used here is $\mathbf{d} = (d, e, f)$. Common direction vectors to use are either the radial direction vector $\mathbf{d} = (t, 1-t, -1)$ or some constant direction vector parallel to either the $\mathbf{b_{003}b_{300}}$ or $\mathbf{b_{003}b_{030}}$ edge, i.e., either $\mathbf{d} = (1, 0, -1)$ or $\mathbf{d} = (0, 1, -1)$.

The directional derivative [5] of a surface $\mathbf{F}$ at the point $\mathbf{u} = (u_0, u_1, u_2)$ and with respect to the vector $\mathbf{d} = (d, e, f)$ is given by

$$D_{\mathbf{d}}\mathbf{F}(\mathbf{u}) = d\mathbf{F_{u_0}}(\mathbf{u}) + e\mathbf{F_{u_1}}(\mathbf{u}) + f\mathbf{F_{u_2}}(\mathbf{u})$$

where $\mathbf{F_{u_0}}, \mathbf{F_{u_1}}$, and $\mathbf{F_{u_2}}$ are the partial derivatives of $\mathbf{F}$ with respect to u_0, u_1, and u_2, i.e.,

$$\mathbf{F_{u_0}} = \frac{\partial}{\partial u_0}\mathbf{F}(\mathbf{u}),\ \mathbf{F_{u_1}} = \frac{\partial}{\partial u_1}\mathbf{F}(\mathbf{u}),\ \mathbf{F_{u_2}} = \frac{\partial}{\partial u_2}\mathbf{F}(\mathbf{u})$$

The patch $\mathbf{F}(\mathbf{u})$ is given by

$$\mathbf{F}(\mathbf{u}) = \sum_{|\vec{i}|=3} B^3_{\vec{i}}(\mathbf{u})\mathbf{c}(\mathbf{u}) = \sum_{|\vec{i}|=3} \frac{3!}{i_0!i_1!i_2!} u_0^{i_0} u_1^{i_1} u_2^{i_2}\mathbf{c}(\mathbf{u})$$

where

$$\begin{aligned}
\mathbf{c_{300}}(\mathbf{u}) &= \mathbf{b_{300}},\ \mathbf{c_{030}}(\mathbf{u}) = \mathbf{b_{030}},\ \mathbf{c_{003}}(\mathbf{u}) = \mathbf{b_{003}} \\
\mathbf{c_{111}}(\mathbf{u}) &= \frac{u_1u_2\mathbf{b_{111,0}} + u_0u_2\mathbf{b_{111,1}} + u_0u_1\mathbf{b_{111,2}}}{u_0u_1 + u_0u_2 + u_1u_2} \\
\mathbf{c}_{\sigma(\mathbf{012})}(\mathbf{u}) &= h_{\sigma_0\sigma_1}(\mathbf{u})\mathbf{b}_{\sigma(\mathbf{012}),\sigma_\mathbf{0}} + h_{\sigma_1\sigma_0}(\mathbf{u})\mathbf{b}_{\sigma(\mathbf{012}),\sigma_\mathbf{1}}
\end{aligned}$$

The evaluation can be simplified by recognizing that $u_2 = 0$ when calculating the cross boundary derivative along the $\mathbf{b_{300}b_{030}}$ edge. Any terms with u_2 still left as a factor after taking the partial derivative can be discarded. Each of the following expressions for $\mathbf{F_{u_0}}(\mathbf{u})$, $\mathbf{F_{u_1}}(\mathbf{u})$, and $\mathbf{F_{u_2}}(\mathbf{u})$ leave out terms which have a factor of u_2 after differentiation.

$$\begin{aligned}
\mathbf{F_{u_0}}(\mathbf{u}) \mid_{u_2=0} &= \frac{\partial}{\partial u_0}\left[B^3_{300}(\mathbf{u})\mathbf{c}_{300}(\mathbf{u}) + B^3_{210}(\mathbf{u})\mathbf{c}_{210}(\mathbf{u}) + B^3_{120}(\mathbf{u})\mathbf{c}_{120}(\mathbf{u})\right]_{u_2=0} \qquad (4.4) \\
&= 3\left[B^2_{200}(\mathbf{u})\mathbf{b_{300}} + B^2_{110}(\mathbf{u})\mathbf{b_{210,2}} + B^2_{020}(\mathbf{u})\mathbf{b_{120,2}}\right]_{u_2=0}
\end{aligned}$$

$$\mathbf{F_{u_1}}(\mathbf{u}) \mid_{u_2=0} = \frac{\partial}{\partial u_1}\left[B^3_{210}(\mathbf{u})\mathbf{c}_{210}(\mathbf{u}) + B^3_{120}(\mathbf{u})\mathbf{c}_{120}(\mathbf{u}) + B^3_{030}(\mathbf{u})\mathbf{c}_{030}(\mathbf{u})\right]_{u_2=0} \qquad (4.5)$$

$$= 3\left[B^2_{200}(\mathbf{u})\mathbf{b_{210,2}} + B^2_{110}(\mathbf{u})\mathbf{b_{120,2}} + B^2_{020}(\mathbf{u})\mathbf{b_{030}}\right]_{u_2=0}$$

$$\mathbf{F_{u_2}}(\mathbf{u})\ |_{u_2=0} = \frac{\partial}{\partial u_2}\left[B^3_{201}(\mathbf{u})\mathbf{c}_{201}(\mathbf{u}) + B^3_{111}(\mathbf{u})\mathbf{c}_{111}(\mathbf{u}) + B^3_{021}(\mathbf{u})\mathbf{c}_{021}(\mathbf{u})\right]_{u_2=0} \tag{4.6}$$

$$= 3\left[B^2_{200}(\mathbf{u})\mathbf{b_{201,2}} + B^2_{110}(\mathbf{u})\mathbf{b_{111,2}} + B^2_{020}(\mathbf{u})\mathbf{b_{021,2}}\right]_{u_2=0}$$

The resulting directional derivative is

$$\begin{aligned} D_{(d,e,f)}\mathbf{F}(t, 1-t, 0) = {} & [d\mathbf{b_{300}} + e\mathbf{b_{210,2}} + f\mathbf{b_{201,2}}]B^2_{200}(t, 1-t, 0) + \\ & [d\mathbf{b_{210,2}} + e\mathbf{b_{120,2}} + f\mathbf{b_{111,2}}]B^2_{110}(t, 1-t, 0) + \\ & [d\mathbf{b_{120,2}} + e\mathbf{b_{030}} + f\mathbf{b_{021,2}}]B^2_{020}(t, 1-t, 0) \end{aligned} \tag{4.7}$$

The directional derivative [5] of a cubic triangular Bézier patch, with control points $\{\mathbf{b_{300}}, \mathbf{b_{030}}, \mathbf{b_{003}}, \mathbf{b_{\vec{i},2}}\}$, at $\mathbf{u} = (t, 1-t, 0), t \in (0, 1)$, is identical to Equation 4.7. The cross boundary derivative on the $\mathbf{b_{300}}\mathbf{b_{030}}$ edge is given by substituting $\mathbf{d} = (d, -d, 0), d \neq 0$. By symmetry in the structure of the patch formula, the cross boundary derivative along the other two boundaries has a form similar to Equation 4.7.

The patch can be used to construct a G^1 network of parametric triangular patches since rational blends solve the vertex consistency problem by combining independently constructed boundary tangent plane fields. However, the patch provides more than independent boundary tangent plane fields. The Foley-Opitz scheme uses a convex combination of three triangular Bézier patches that must share the same boundary – they only differ in the interior control point. This patch gives the freedom to choose the boundaries independently for each of the three Bézier patches. Two neighbouring patches can be considered locally functional patches by choosing some plane as a natural parameterization. Functional interpolation techniques can then be used to construct a G^1 join. This process can be repeated for each patch pair to give a G^1 network.

Choosing a Plane

The relationship between two patches is defined by choosing a plane with respect to which the patches can temporarily be treated as functional patches. Using a plane in the range space

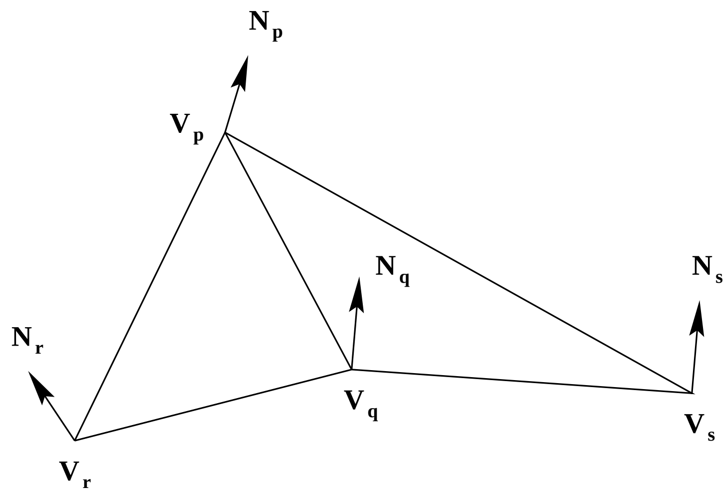

Figure 4.8: Two neighbouring data triangles with vertices $\mathbf{V_p}$, $\mathbf{V_q}$, $\mathbf{V_r}$, and $\mathbf{V_s}$, and normals $\hat{\mathbf{N}}_\mathbf{p}$, $\hat{\mathbf{N}}_\mathbf{q}$, $\hat{\mathbf{N}}_\mathbf{r}$, and $\hat{\mathbf{N}}_\mathbf{s}$.

as a temporary "natural parameterization" for the two patches allows easy construction of both boundaries and C^1 joins between the two patches. To construct a network of patches, a different plane must be chosen for every triangle boundary. Let the two triangles that we are trying to join together, shown in Figure 4.8, have vertices

$$\mathbf{V_p},\ \mathbf{V_q},\ \mathbf{V_r},\ \mathbf{V_s}$$

and normals

$$\hat{\mathbf{N}}_\mathbf{p},\ \hat{\mathbf{N}}_\mathbf{q},\ \hat{\mathbf{N}}_\mathbf{r},\ \hat{\mathbf{N}}_\mathbf{s}$$

For the $\mathbf{V_p V_q}$ boundary some plane must be chosen to facilitate the Foley-Opitz construction. The obvious restriction that would apply to such a plane is that the projections of the two triangles onto the plane cannot overlap and that neither of the triangles can be projected onto a line segment. Figure 4.9 illustrates these restrictions.

Note that what is important here is not the position of the plane but only its orientation relative to the two triangles. Thus only a normal is needed to characterize the plane since whenever height values are required any point can be substituted into the point-normal definition of the plane. Let $\hat{\mathbf{N}}$ be the normal we are trying to select.

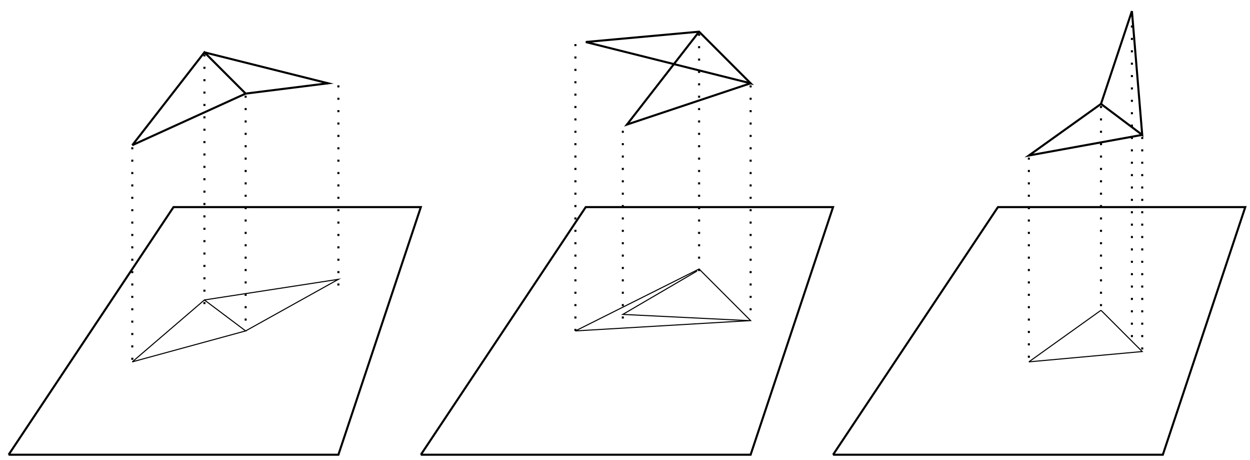

Figure 4.9: The three cases for projecting triangle pairs: acceptable projection, left, overlapping domains, middle, degenerate projection, right.

The restriction on the plane relating to projections of the two triangles is that

$$(\vec{\mathbf{C}}_{\mathbf{pr}} \cdot \hat{\mathbf{N}})(\vec{\mathbf{C}}_{\mathbf{sp}} \cdot \hat{\mathbf{N}}) > 0 \tag{4.8}$$

where

$$\vec{\mathbf{C}}_{\mathbf{pr}} = \vec{\mathbf{V}}_{\mathbf{pq}} \times \vec{\mathbf{V}}_{\mathbf{rq}},\ \vec{\mathbf{C}}_{\mathbf{sp}} = \vec{\mathbf{V}}_{\mathbf{sq}} \times \vec{\mathbf{V}}_{\mathbf{pq}},$$

and

$$\vec{\mathbf{V}}_{\mathbf{pq}} = \mathbf{V_p} - \mathbf{V_q}\vec{\mathbf{V}}_{\mathbf{rq}} = \mathbf{V_r} - \mathbf{V_q}\vec{\mathbf{V}}_{\mathbf{sq}} = \mathbf{V_s} - \mathbf{V_q}$$

I have chosen two different methods for selecting a plane. The first method is to take the average of the normals at the two triangle vertices on the common border between the two patches.

$$\hat{\mathbf{N}} = \frac{\hat{\mathbf{N}}_{\mathbf{p}} + \hat{\mathbf{N}}_{\mathbf{q}}}{\mid \hat{\mathbf{N}}_{\mathbf{p}} + \hat{\mathbf{N}}_{\mathbf{q}} \mid}$$

The second is to take the vector that is perpendicular to the common edge between the two triangles and that bisects the two normals defining the two planes containing the two data triangles. The normal is then

$$\hat{\mathbf{N}} = \frac{\hat{\mathbf{N}}_{\mathbf{pqr}} + \hat{\mathbf{N}}_{\mathbf{pqs}}}{\mid \hat{\mathbf{N}}_{\mathbf{pqr}} + \hat{\mathbf{N}}_{\mathbf{pqs}} \mid}$$

where $\hat{\mathbf{N}}_{\mathbf{pqr}}$ and $\hat{\mathbf{N}}_{\mathbf{pqs}}$ are the normals that define the planes containing the two triangles.

The first method does not guarantee that the normal satisfies the criteria given above but does incorporate the information given by the data normals. When a normal is constructed that does not satisfy the criteria, I revert to the second method of choosing a plane.

The second method will not work when the two data triangles lie on the same plane. In this situation the input data will have to be modified.

Boundaries

The construction of a projection plane as described above requires two neighbouring data triangles. However, special treatment is required when constructing patches that border the edge of a data set since these patches will have one or more edges for which no neighbour exists. When constructing a patch associated with one of these bordering edges I use the data triangle as the projection plane.

Using Other Blends

In my formulation of parametric Foley-Opitz patches, I have used the function

$$\mathbf{c_{111}}(\mathbf{u}) = \frac{u_1u_2\mathbf{b_{111,0}} + u_0u_2\mathbf{b_{111,1}} + u_0u_1\mathbf{b_{111,2}}}{u_0u_1 + u_0u_2 + u_1u_2}$$

when blending interior control points. Other weights may be used when blending the interior control points provided that their sum is one. For instance, weights of the form

$$\mathbf{c_{111}}(\mathbf{u}) = \frac{u_1^nu_2^n\mathbf{b_{111,0}} + u_0^nu_2^n\mathbf{b_{111,1}} + u_0^nu_1^n\mathbf{b_{111,2}}}{u_0^nu_1^n + u_0^nu_2^n + u_1^nu_2^n}$$

where n is an integer can be used since

$$\sum \frac{u_i^nu_j^n}{u_0^nu_1^n + u_0^nu_2^n + u_1^nu_2^n} = 1$$

However, in practice, increasing the exponent of each parameter did not improve surface quality. Variations on the weights for the boundary control points did not give any improvement over the weights I have already outlined.

The Foley-Opitz functional construction produces patches with cubic precision by using information from 9 control points per edge – three rows of control points. In my formulation of parametric Foley-Opitz patches, I have included only the first two rows of control points in the final interpolant. Although three rows were used to set the interior control points, the third row of control points is thrown out. I have experimented with incorporating these points into the blend but the resulting surfaces did not improve on the simpler blend I have already outlined.

Chapter 5

Analysis of the New Scheme

The formulation of parametric Foley-Opitz patches is an interesting variation on existing schemes because the blending of boundary control points allows the three Bézier patches which form a parametric Foley-Opitz patch to be constructed independently – they are not required to share the same boundary, only the same corner points. This allows the incorporation of elements from functional data fitting schemes into neighbouring patch pair constructions.

One consequence of using this rational blend is that removable singularities are introduced at the patch corners. When evaluated at a corner, the term $frac00$ will appear as the weight of the corner control point. In this case, the patch formula should not be used. When evaluating near the corner, attention should be paid to possible numerical instability.

Parametric Foley-Opitz patches have some basic properties that are necessary to establish the scheme as a valid option for data interpolation: parametric Foley-Opitz patches interpolate the given data with G^1 continuity, and the contributions of the three Bézier patches to the three boundary tangent plane fields are independent. In contrast to these mathematical conclusions, the analysis in this chapter will be mostly qualitative and will largely involve evaluation of the visual quality of surfaces.

I have constructed interpolants for three data sets that demonstrate the effect of variations in Foley-Opitz patch construction, evaluate how the surface is affected by the density of the input data, contrast surfaces produced by other interpolation schemes, and isolate different

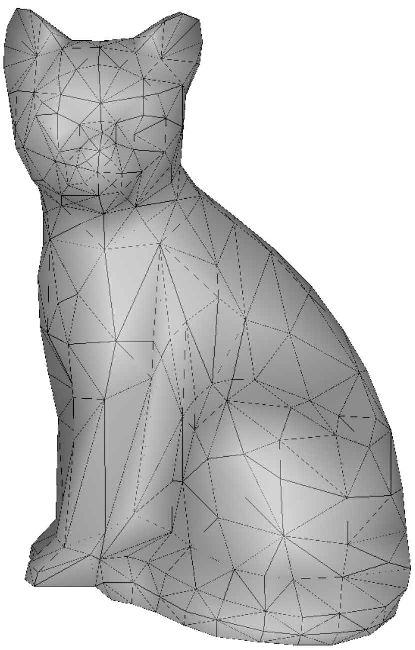

Figure 5.1: Cat test data set.

components of parametric Foley-Opitz patches.

The cat data set, shown in Figure 5.1, will be used as test data for some of the discussion of surface quality and the choice of parameterization plane. The data set includes vertex positions, normals, and a mesh of triangular faces. Each normal in the cat data set was constructed by weighting the surrounding face normals with the inverse area of the faces. Note that this is not the only method of construction of normals from data points and that different techniques could influence the resulting surface shape. There are 366 vertices and 698 triangular faces in the data set.

The two tori data sets, shown in Figure 5.2, contain vertex positions, normals, and second fundamental forms sampled around the torus at regular intervals, plus a mesh of triangular faces. The torus was sampled at regular intervals both around the large circle and around the small circle. The first data set has 10 samples around the large circle and 5 samples around the small circle giving 50 vertices and 100 triangular faces. The 10 x 10 sampled data set has 100 points and 200 faces.

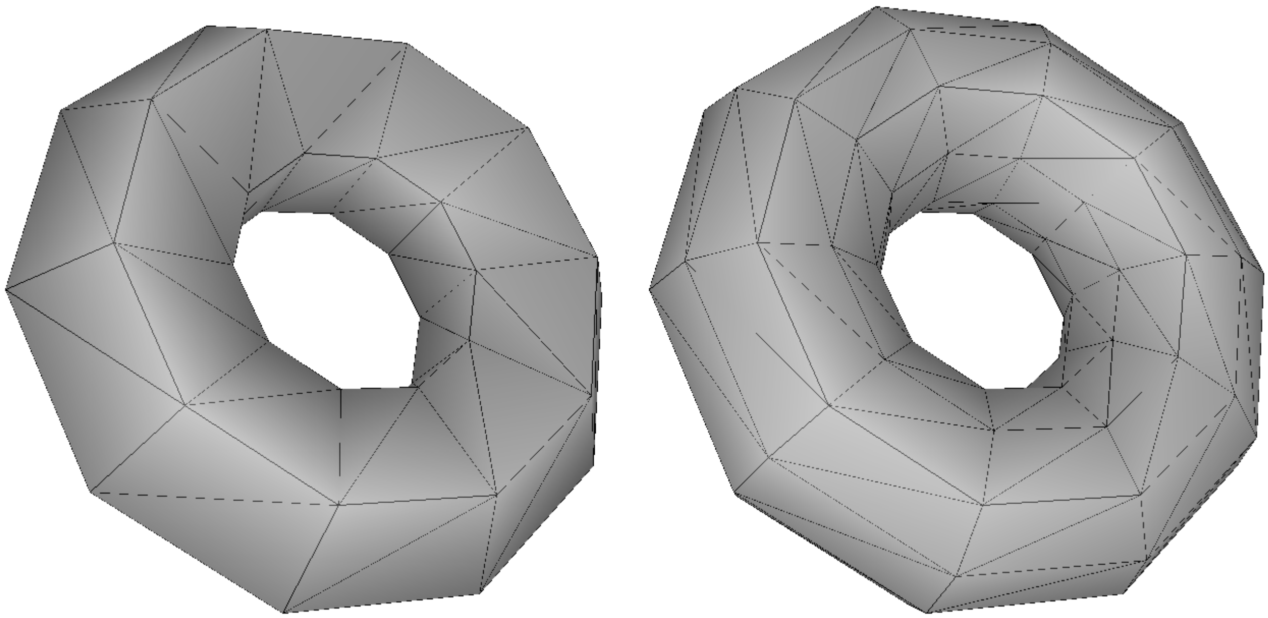

Figure 5.2: Tori data sets sampled 10x5, left, and 10 x 10, right.

5.1 Choosing a Projection Plane

Section 4.2 outlined two methods for choosing the plane that is used as a local parameterization between two neighbouring triangles – using the plane perpendicular to the bisecting plane of the two triangles that contains the common edge and using the plane defined by the average of the normals from the common edge. The first uses only the information provided by the input points while the second uses only two input normals.

Using the plane determined by the bisecting plane guarantees that the construction can be done since there will be no overlapping projections of triangle pairs. Using this plane means that parametric Foley-Opitz patches can then be constructed for any data, provided that the normals are not parallel to the plane of projection. This plane is the natural choice and at first inspection, the superior choice. However, after examining some constructed surfaces I found this method to be of limited use.

As a rough measure of the quality of patches produced by the parametric the Foley-Opitz scheme I have chosen to consider those patches that have control points that are "far" from

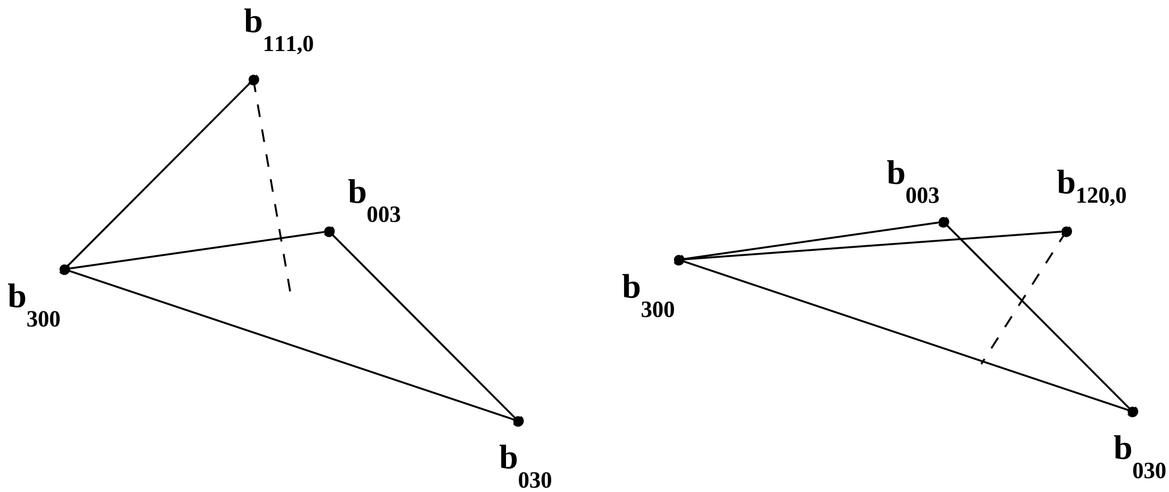

Figure 5.3: Some conditions for good patches: $| \mathbf{b_{111,0}} - \mathbf{b_{300}} |< C_i | \mathbf{b_{003}} - \mathbf{b_{300}} |$ or $| \mathbf{b_{111,0}} - \mathbf{b_{300}} |< C_i | \mathbf{b_{030}} - \mathbf{b_{300}} |$, $| \mathbf{b_{120,0}} - \mathbf{b_{300}} |< C_b | \mathbf{b_{030}} - \mathbf{b_{300}} |$.

the underlying data triangle to be unacceptable. By "far," I mean that the distance between boundary control points, interior control points and the corner control points exceeds some tolerance. For example, Figure 5.3 shows a data triangle plus the $\mathbf{b_{111,0}}$ and $\mathbf{b_{120,0}}$ control points. $\mathbf{b_{111,0}}$ would be an unacceptable control point if

$$| \mathbf{b_{111,0}} - \mathbf{b_{300}} |> C_i | \mathbf{b_{030}} - \mathbf{b_{300}} |$$

and

$$| \mathbf{b_{111,0}} - \mathbf{b_{300}} |> C_i | \mathbf{b_{003}} - \mathbf{b_{300}} | \tag{5.1}$$

$\mathbf{b_{120,0}}$ would be unacceptable if

$$| \mathbf{b_{120,0}} - \mathbf{b_{300}} |< C_b | \mathbf{b_{030}} - \mathbf{b_{300}} | \tag{5.2}$$

C_i and C_b are constants that represent tolerances for the interior and boundary conditions respectively. If any one control does not satisfy the conditions then the whole patch is treated as unacceptable. Note that interior points are compared against two triangle sides while boundary points are compared against only one side.

Figure 5.4 shows the parametric Foley-Opitz scheme run on the cat data set with patches removed that do not satisfy the conditions given by Equations 5.1 and 5.2 with tolerances

Figure 5.4: Parametric Foley-Opitz patches fitted to the cat data set using a projection plane that is perpendicular to the bisecting plane. Patches failing to meet the tolerance $C_i = 3$ and $C_b = 3$ were not rendered. 76 of 698 patches failed to meet the tolerance conditions.

of $C_i = 3$ and $C_b = 3$. 76 of the 698 patches were not rendered since they did not satisfy the tolerance conditions. The large number of rejected patches is the result of the lack of correlation between the projection plane and the vertex normals.

The patches produced by using the plane defined by the average of two normals are of much higher quality than those produced using the bisecting plane method. Figure 5.5 shows the complete interpolant. The only areas where the interpolation produces inadequate patches are on the ears of the cat data set. The rest of the interpolant preserved much of the underlying surface from which the data was sampled. I pruned patches using three different threshold values to see if any patches other than those on the cat's ears would be removed. Rejection rates for the three interpolants in Figure 5.6 are: 4 of 698 patches rejected with tolerance settings of $C_i = 3$ and $C_b = 3$, 8 of 698 patches rejected with tolerance settings of $C_i = 2$ and $C_b = 2$, and 21 of 698 patches rejected with tolerance settings $C_i = 1.5$ and

Figure 5.5: Average of normals method used to interpolate the cat data set.

$C_b = 1.5$. The only patches removed were from the cat's ears.

The explanation for the superior surface quality lies in the relationship of the two normals to the Bézier patch boundary control points. The average of normals will ideally produce boundary curves that are parallel to the projection plane at the two triangle vertices on the shared boundary. This means that the boundary curve along the shared edge will have little variation from the data triangle. The other two boundary curves will also have little variation near the shared edge. Of course, these boundary curve properties break down if the two vertices along the shared edge are sampled from the opposite sides of a crease in the underlying data set.

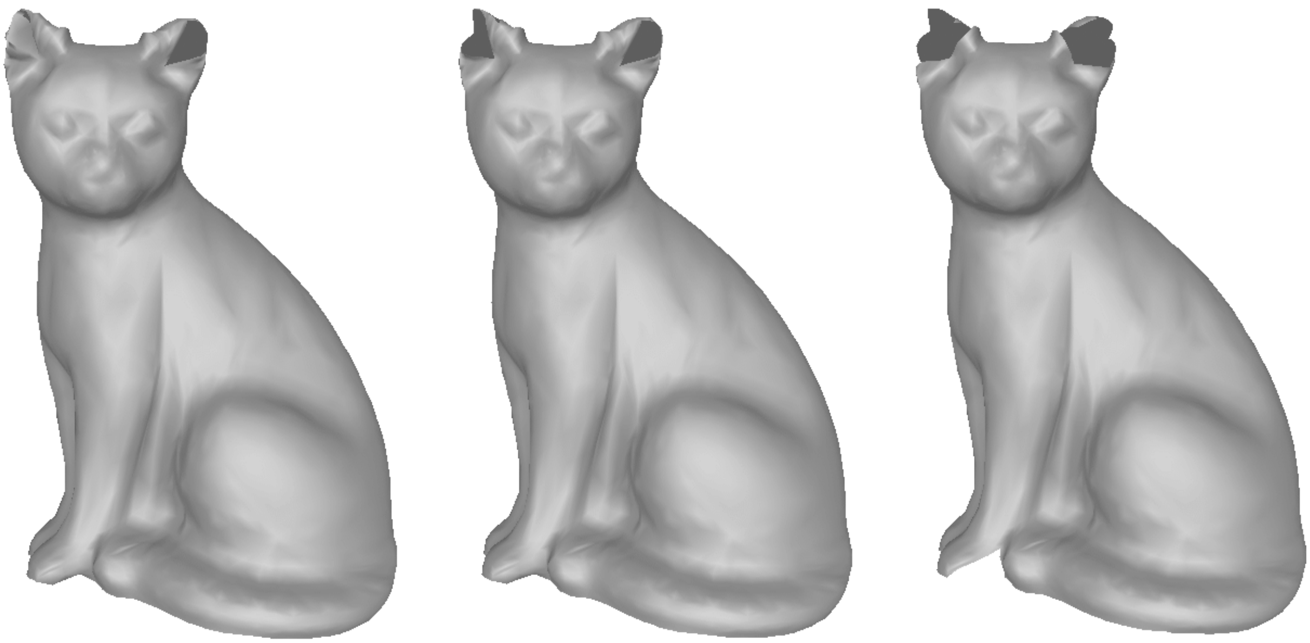

Figure 5.6: Tolerance settings of $C_i, C_b = 3$, $C_i, C_b = 2$, and $C_i, C_b = 1.5$ with average of normals method.

The remaining boundary points, near the third vertex that is not on the shared edge, contribute to the construction of the interior point but do not appear in the final patch blend so they play a less important role than the control points near the shared edge. The importance of the control points near the shared edge combined with their dependency on the two normals make the average of normals a good choice for the parameterization plane. Using the plane perpendicular to the bisecting plane does not use the information provided by the normals possibly ignoring some useful information.

When used to interpolate data sampled from the torus both methods of choosing a plane produced reasonable patches. Figure 5.7 presents a comparison of the two methods on the 10 x 5 torus data set. The surface produced by using the bisecting plane method is obviously better although some creases can be seen that correspond to patch boundaries. The other surface has noticeable, regular bumps that are more serious defects than the other interpolant.

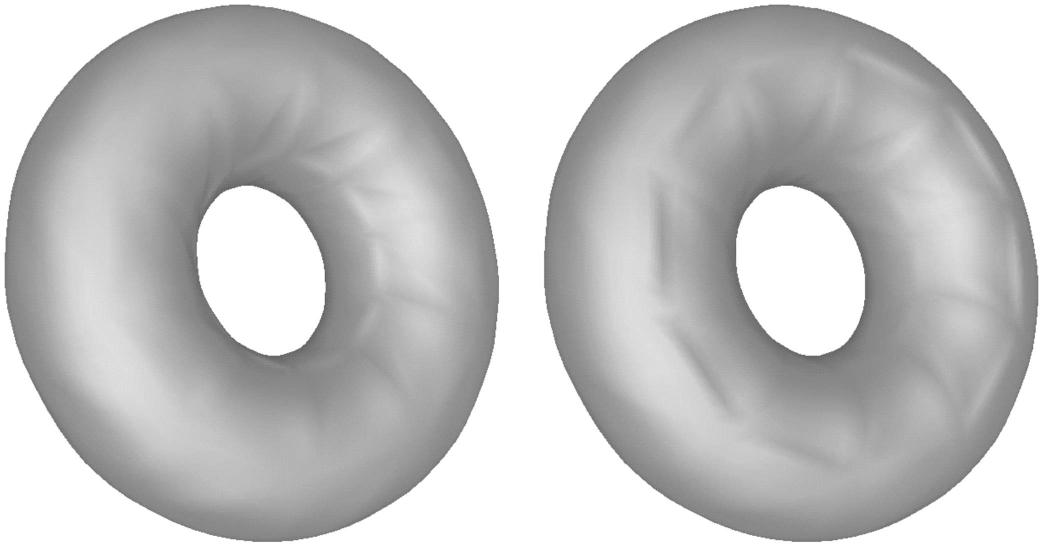

Figure 5.7: Parametric Foley-Opitz interpolants for 10 x 5 sampled torus data set using projection planes perpendicular to the bisecting plane, left, and taken from the average of normals, right.

5.2 Data Sampling Frequency

The comparison between the two methods of choosing a projection plane using the torus data set produced minor differences but comparisons between data sets of various sampling frequencies give some interesting results.

Figure 5.8 shows the parametric Foley-Opitz interpolants for the two torus data sets using the average of normals to define the projection plane. Not only is an extremely good surface produced by doubling the samples taken from the torus but there is dramatic improvement in comparison to the interpolant for the coarser data set.

Any reasonable interpolation scheme will produce better surfaces with more finely sampled data so the improvement with Foley-Opitz patches was expected. However, the rate of convergence of the parametric Foley-Opitz interpolant to the torus was unexpected. The next section will contrast the rates of improvement of parametric Foley-Opitz patches and some representative parametric patches.

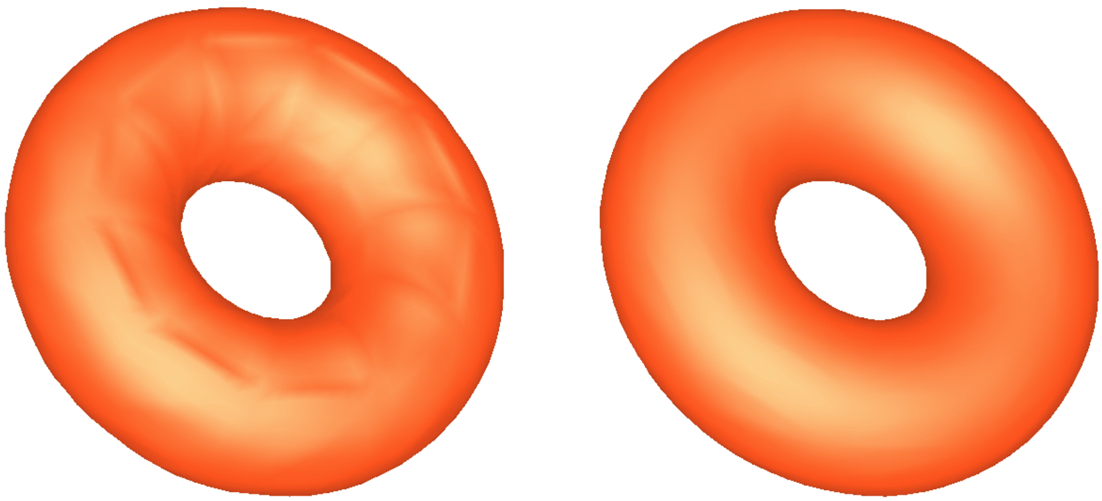

Figure 5.8: Parametric Foley-Opitz interpolants using average of normals for 10x5, left, and 10 x 10, right, torus data sets.

5.3 Comparison With Other Parametric Schemes

I will use the average of normals method to choose the projection plane for parametric Foley-Opitz patches. Comparing the reasonable results given by the average of normals method to the mixed results given by the bisecting plane method makes the former the better general purpose method.

I will compare parametric Foley-Opitz interpolants with Shirman-Sequin and triangular Gregory patches. Mann [16] has concluded that existing parametric local triangular interpolation schemes produce surfaces of similar quality, so I will consider Shirman-Sequin and triangular Gregory patches as representative of split domain and rational blend schemes respectively.

The cubic boundary curves I used with these two schemes are constructed from vertex positions and normals by first choosing a plane then intersecting it with the tangent planes at the two vertices. The two middle control points lie on the lines of intersection and are chosen so that the distance from one of the vertices is one third the distance between the

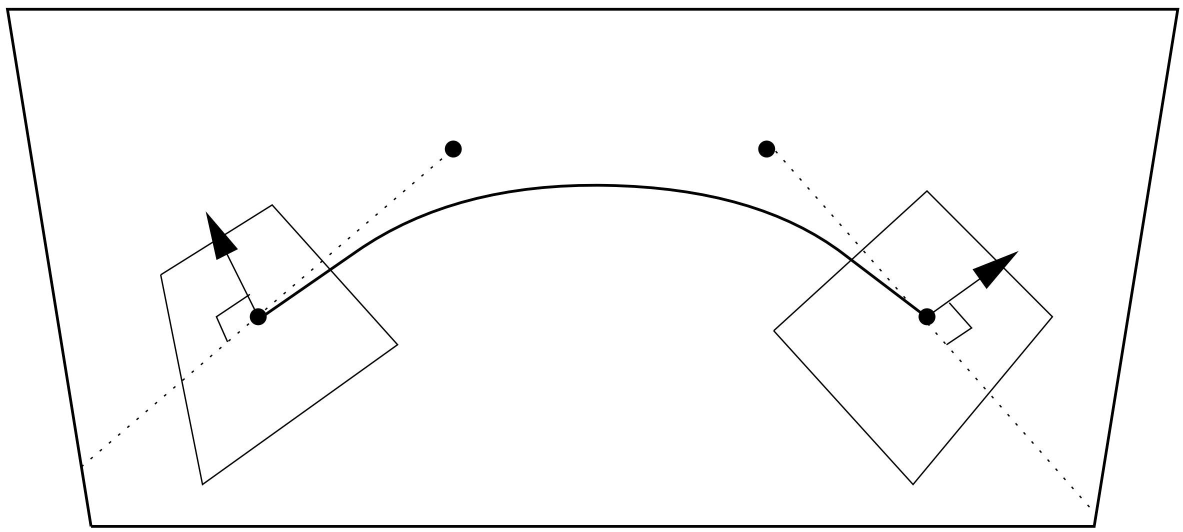

Figure 5.9: Planar boundary curve. Each of the two interior control points lies on the tangent plane of an endpoint. The distance between an interior control point and one of the endpoints is one third the distance between the two endpoints. All four control points are coplanar.

two vertices, shown in Figure 5.9.

Additionally, I constructed de Boor-Hollig-Sabin boundary curves with the two tori data sets to see how the additional information provided by the second fundamental forms would affect the surface quality. de Boor-Hollig-Sabin curves are cubic curves that interpolate vertex positions, normals, and curvature information [4].

Figure 5.10 shows the cat data set interpolated with Shirman-Sequin, triangular Gregory patches, and parametric Foley-Opitz patches, using planar boundaries for the first two schemes, and the average of normals method for the third scheme. Except for the pruned defects on the ears, the parametric Foley-Opitz interpolant is a better surface. The other two methods preserve evidence of the underlying geometry of the patches – patch boundaries can be seen on the surfaces in many places. All three surfaces are smooth G^1 surfaces but many of the surface defects from the first two interpolants have been smoothed out in the parametric Foley-Opitz interpolant, though the surface is still far from perfect.

Figure 5.11 shows the 10 x 5 sampled torus data set interpolated with all three schemes, using the average of normals method for parametric Foley-Opitz patches and planar boundary

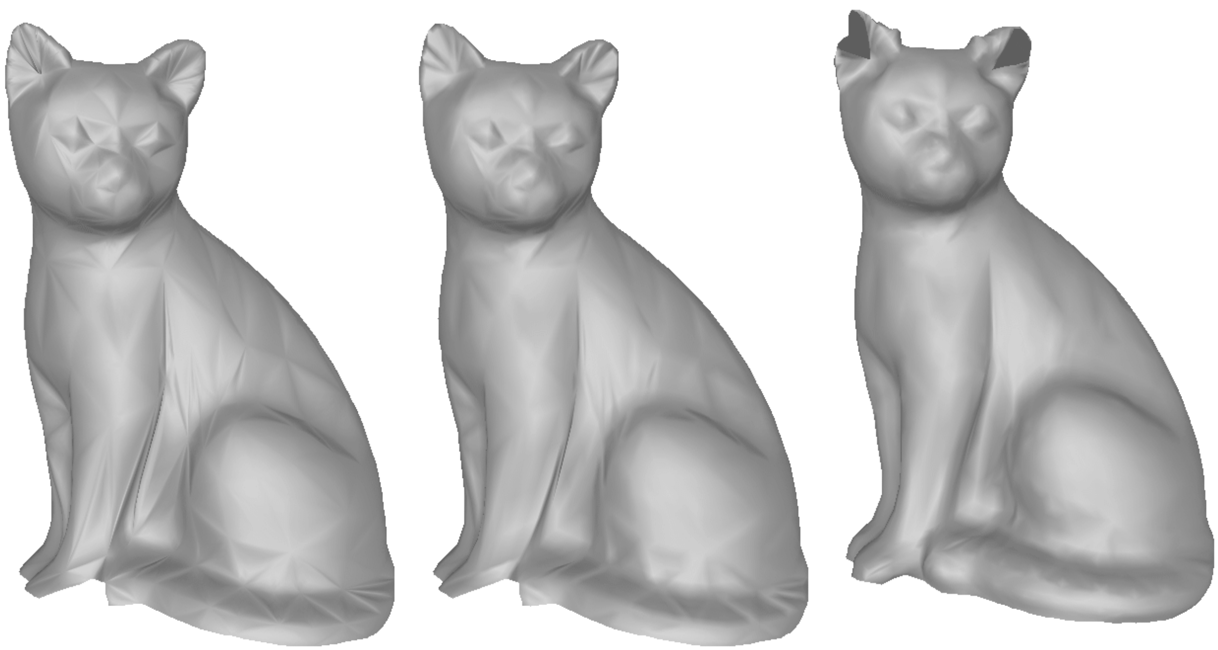

Figure 5.10: Comparison of Shirman-Sequin, triangular Gregory patches, and parametric Foley-Opitz patches on the cat data set using planar boundary curves with Shirman-Sequin and triangular Gregory patches.

Figure 5.11: Comparison of Shirman-Sequin, triangular Gregory patches, and parametric Foley-Opitz patches on the 10 x 5 torus data set using planar boundary curves with Shirman-Sequin and triangular Gregory patches.

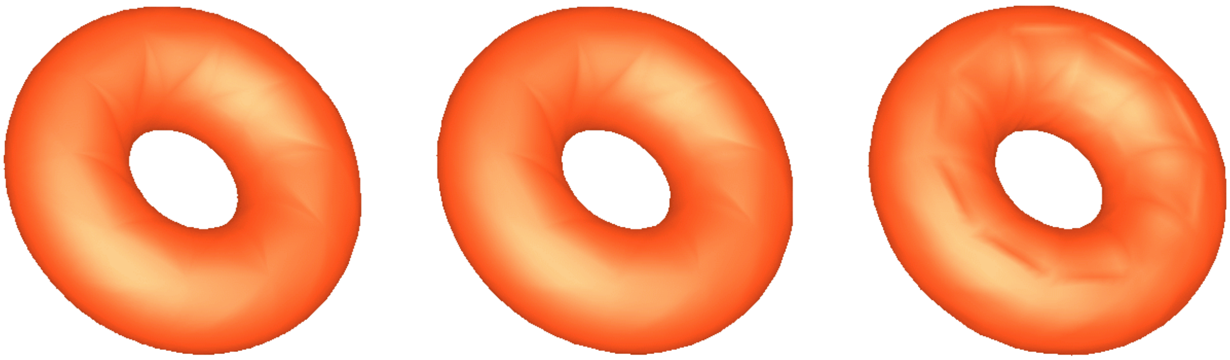

Figure 5.12: Comparison of Shirman-Sequin, triangular Gregory patches, and parametric Foley-Opitz patches on the 10 x 5 torus data set using de Boor-Hollig-Sabin boundary curves with Shirman-Sequin and triangular Gregory patches.

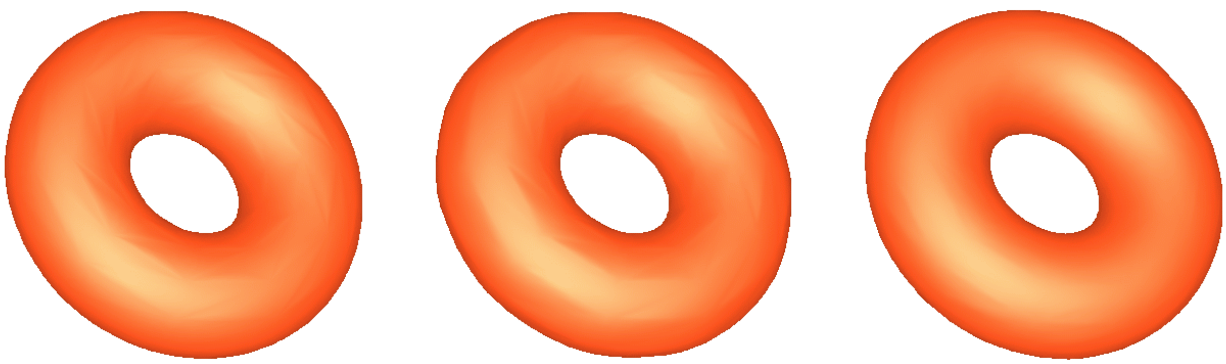

Figure 5.13: Shirman-Sequin, left, triangular Gregory patches, middle, and parametric Foley-Opitz patches, right, used to interpolate the 10 x 10 torus data set. Planar boundaries were used with Shirman-Sequin and triangular Gregory patches.

curves for Shirman-Sequin and triangular Gregory patches. Both of the other two schemes produce better surfaces than the parametric Foley-Opitz surface. There are noticeable lumps with parametric Foley-Opitz. However, evidence of patch boundaries can be seen with both of the other schemes. At this granularity of data sampling, Shirman-Sequin and triangular Gregory patches can be greatly improved by using a different boundary curve construction, de Boor-Hollig-Sabin as in Figure 5.12.

The situation changes greatly in the more densely sampled 10 x 10 torus data set. Fig-

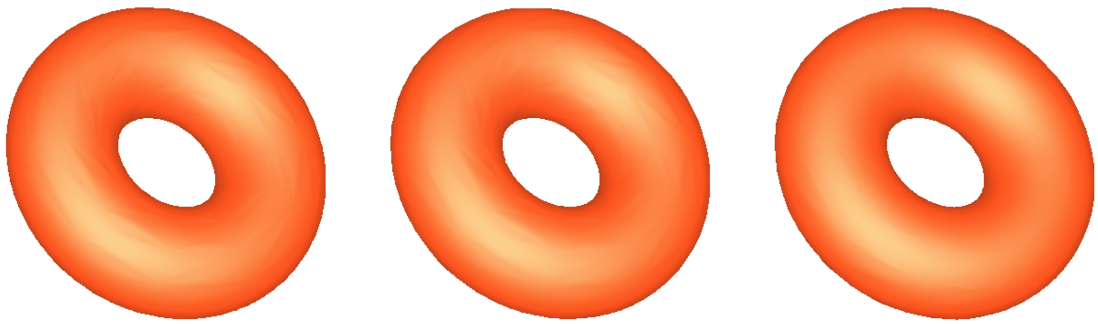

Figure 5.14: Shirman-Sequin, left, triangular Gregory patches, middle, and parametric Foley-Opitz patches, right, used to interpolate the 10 x 10 torus data set. de Boor-Hollig-Sabin boundaries were used with Shirman-Sequin and triangular Gregory patches.

ure 5.13 shows all three schemes with parametric Foley-Opitz patches using the average of normals method and the other schemes using planar boundaries. Parametric Foley-Opitz produces a high quality surface while the other two schemes again produce surfaces with distinctly visible patch boundaries although there is significant improvement over the surface quality produced with the coarser data set.

Figure 5.14 shows all three schemes with parametric Foley-Opitz patches using the average of normals method and the other schemes using de Boor-Hollig-Sabin boundaries. The parametric Foley-Opitz surface is the same as in the previous figure, only the interpolants produced by the other two schemes have changed. Although the superior curve construction improves the smoothness of the other two surfaces, there are still distinct surface defects. de Boor-Hollig-Sabin boundaries use the normals plus the second fundamental forms of the data set while parametric Foley-Opitz patches use only the normals, yet parametric Foley-Opitz produces a better interpolant for this data.

What is interesting about the comparisons using the tori data sets is the significant improvement of parametric Foley-Opitz interpolants when changing from a coarser to a finer sampling of data. Also interesting is the rate of improvement relative to the other two schemes. Parametric Foley-Opitz converges to the original surface much faster than the other schemes.

5.4 Boundary Curves

The improvement in surface quality with the cat data set and the 10 x 10 torus can be explained by analyzing separately the two major components of patch construction, boundary curve construction and tangent plane field construction, to determine which is responsible or if both are equally important. Parametric Foley-Opitz patches cannot be constructed by using the planar or de Boor-Hollig-Sabin boundaries since the projection of curves onto a parameterizing plane is not possible. The only comparison that can be made is substituting the parametric Foley-Opitz boundary curve construction into other schemes.

I have constructed triangular Gregory patches using the curves from parametric Foley-Opitz patches created by the average of normals method. Figure 5.15 shows the cat data set interpolated by triangular Gregory patches, using planar and parametric Foley-Opitz boundaries, and interpolated by parametric Foley-Opitz patches. The problems with patches being fitted around the ears occurs in the triangular Gregory interpolant when using parametric Foley-Opitz boundary curves. Other than that one problem, the rest of the interpolant resembles the original triangular Gregory interpolant more than the parametric Foley-Opitz interpolant. With this data set, the new boundary curves do not improve the Gregory interpolant and actually introduce problems when fitting patches to the ears.

Figure 5.16 and Figure 5.17 show comparisons between triangular Gregory patches using planar boundaries and using parametric Foley-Opitz boundaries with both the coarse and fine torus data sets. Combining the new boundary curves with triangular Gregory patches does significantly alter the surface quality. While the problem of visible patch boundaries is evident the patches are not as flat as those produced using the planar boundaries. The 10 x 10 torus data set surfaces also show marked improvement.

These mixed results – no improvement for the cat data set but marked improvement for the tori data sets – demonstrate that parametric Foley-Opitz boundary curve construction does play an important role in surface quality but does not necessarily produce good surface quality. Unfortunately, in the case of the cat data set, parametric Foley-Opitz patches cannot be improved by substituting in some other parametric curve construction, such as de Boor-Hollig-Sabin curves, because of the locally functional construction.

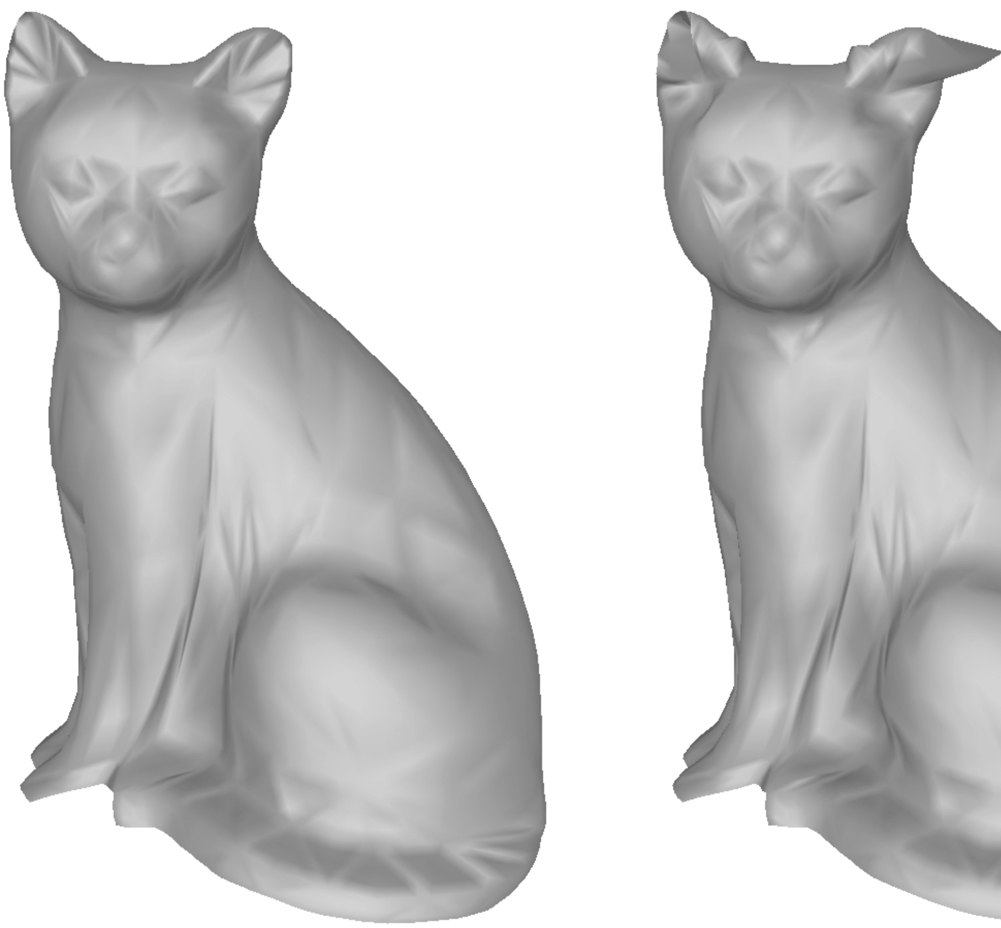

Figure 5.15: Triangular Gregory patches constructed using both planar boundaries, left, and parametric Foley-Opitz boundaries, right, with the cat data set.

5.5 Curvature

So far I have presented images that depict the surfaces using Gouraud shading and Lambertian lighting. Another useful representation of the surface is the curvature plot in which a surface is shaded according to its Gaussian curvature. Curvature plots help highlight areas of both uniform and varying curvature that may be difficult to pick out on the previous images.

The curvature plots in this section render the surface using the numerically estimated

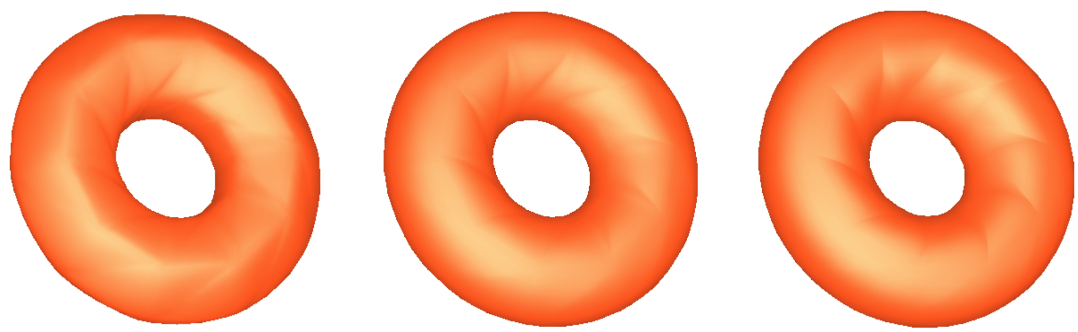

Figure 5.16: Triangular Gregory patches constructed using planar boundaries, left, de Boor-Hollig-Sabin boundaries, middle, and parametric Foley-Opitz boundaries, right, with the 10 x 5 torus data set.

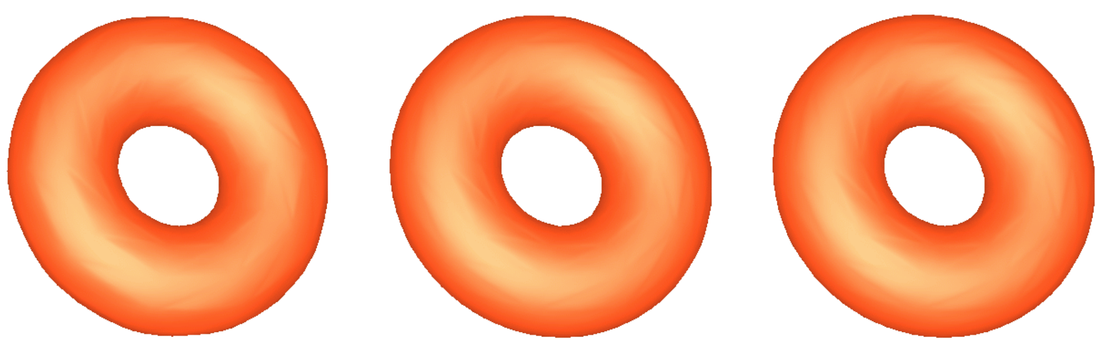

Figure 5.17: Triangular Gregory patches constructed using both planar boundaries, left, de Boor-Hollig-Sabin boundaries, middle, and parametric Foley-Opitz boundaries, right, with the 10 x 10 torus data set.

value of the Gaussian curvature for shading. Gaussian curvature is represented using colour ranging from red, high positive Gaussian curvature, to blue, low negative Gaussian curvature.

The torus will ideally have blue shading along the inner ring, progressing to cyan then green, and reaching yellow along the outer ring, shown in Figure 5.18.

The curvature plots of the torus data sets emphasized the results observed so far – parametric Foley-Opitz patches perform more poorly on the torus 10 x 5 data set and much

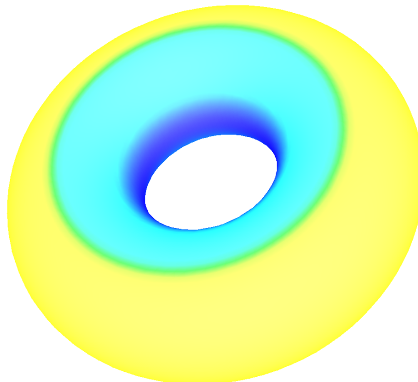

Figure 5.18: Curvature plot for a torus.

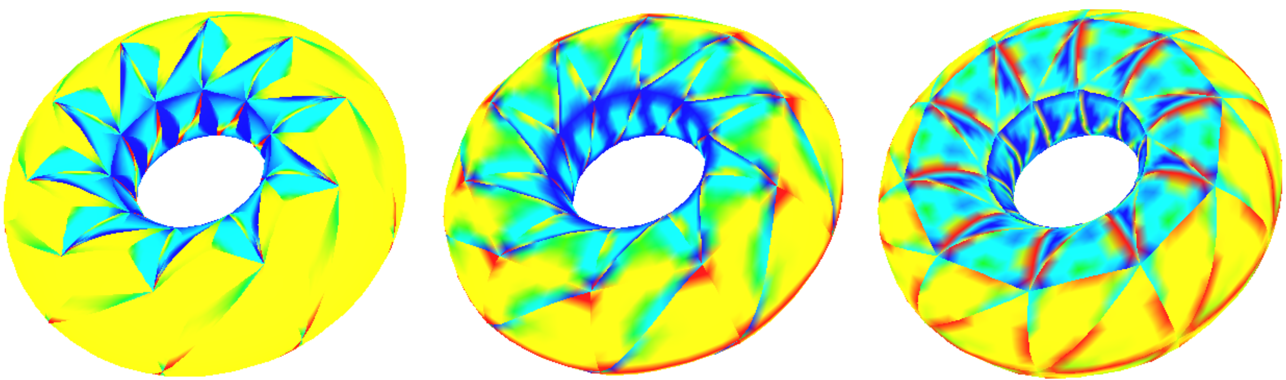

Figure 5.19: Curvature plots for Shirman-Sequin, triangular Gregory patches, and parametric Foley-Opitz patches, with the 10 x 5 torus data set. Planar boundaries were used with Shirman-Sequin and triangular Gregory patches.

better on the torus 10 x 10 data set. Figures 5.19 demonstrates that the bulges in the parametric Foley-Opitz interpolant with the coarse torus data set are visible as areas of high positive curvature, shown as red regions and streaks, on the outer circle. Figure 5.20 shows the fairly uniformly varying curvature plot of the parametric Foley-Opitz interpolant for the finer torus data set.

The curvature plots for the cat data set were inconclusive. The irregularity of the data set and consequently the frequent variation in curvature obscured the differences in curvature between the various interpolation schemes.

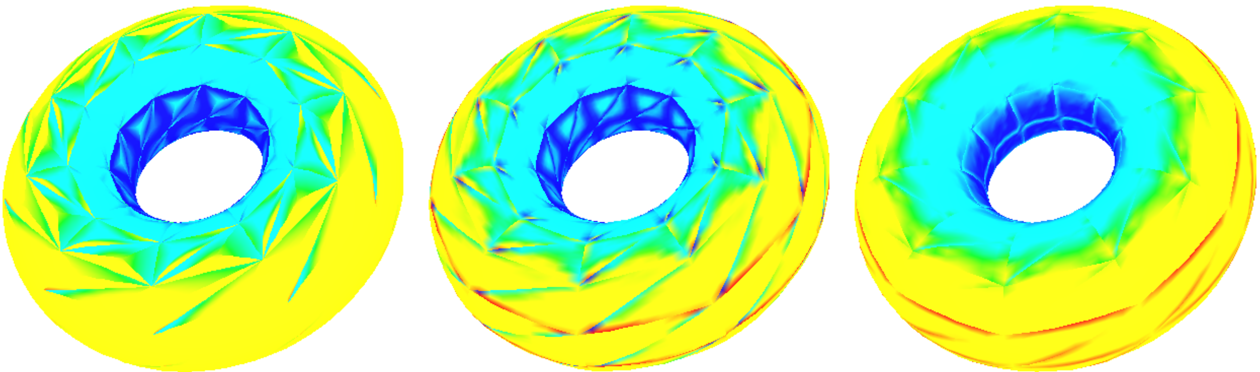

Figure 5.20: Curvature plots for Shirman-Sequin, triangular Gregory patches, and parametric Foley-Opitz patches, with the 10 x 10 torus data set. Planar boundaries were used with Shirman-Sequin and triangular Gregory patches.

5.6 Cross Boundary Derivatives

Since parametric Foley-Opitz boundary curves are not the deciding factor for the improvement in surface quality for either the irregularly sampled data set or the finely tessellated, regularly sampled data set, I have investigated the other major component of triangular patch construction, cross boundary construction.

The surface quality of Foley-Opitz patches results from the cubic precision achieved through the Foley-Opitz cross boundary construction. A corollary to the cubic precision of Foley-Opitz patches are quadratically varying cross boundary derivatives. Parametric Foley-Opitz patches use the same cross boundary construction as functional Foley-Opitz patches giving them quadratically varying derivatives. I have constructed parametric Foley-Opitz interpolants for the three data sets using both the Foley-Opitz cross boundary construction and the cross boundary construction used in Clough-Tocher that gives linearly varying derivatives. Parametric Foley-Opitz boundary curves were used for all test cases.

The cat data set interpolants, shown in Figure 5.21, demonstrate that using linear cross boundary derivatives gives surfaces quite similar to Shirman-Sequin or triangular Gregory patches.

The interpolants for the 10 x 5 torus data set, shown in Figure 5.22, give similar results.

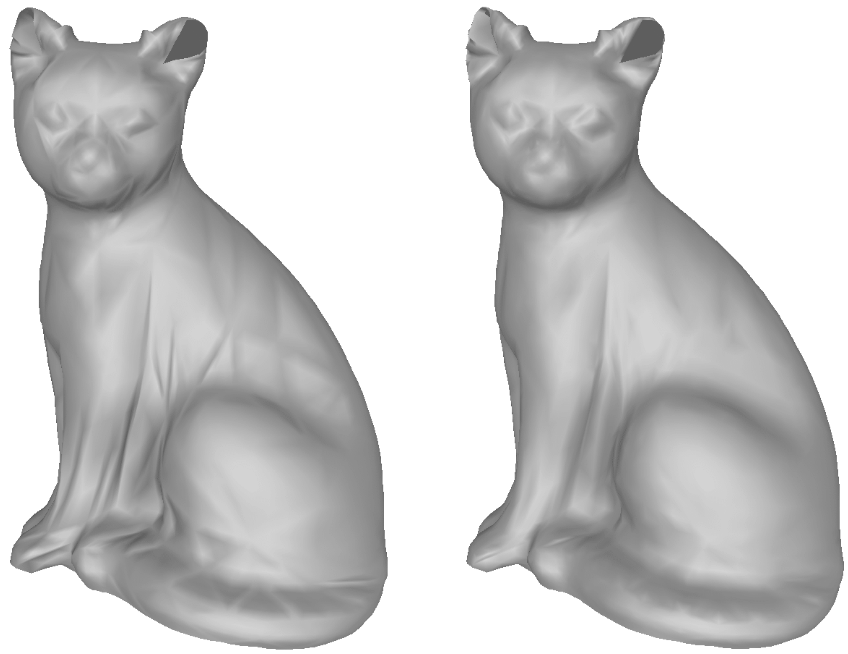

Figure 5.21: The cat data set interpolated by parametric Foley-Opitz boundaries with Clough-Tocher cross boundaries, left, and parametric Foley-Opitz patches, right.

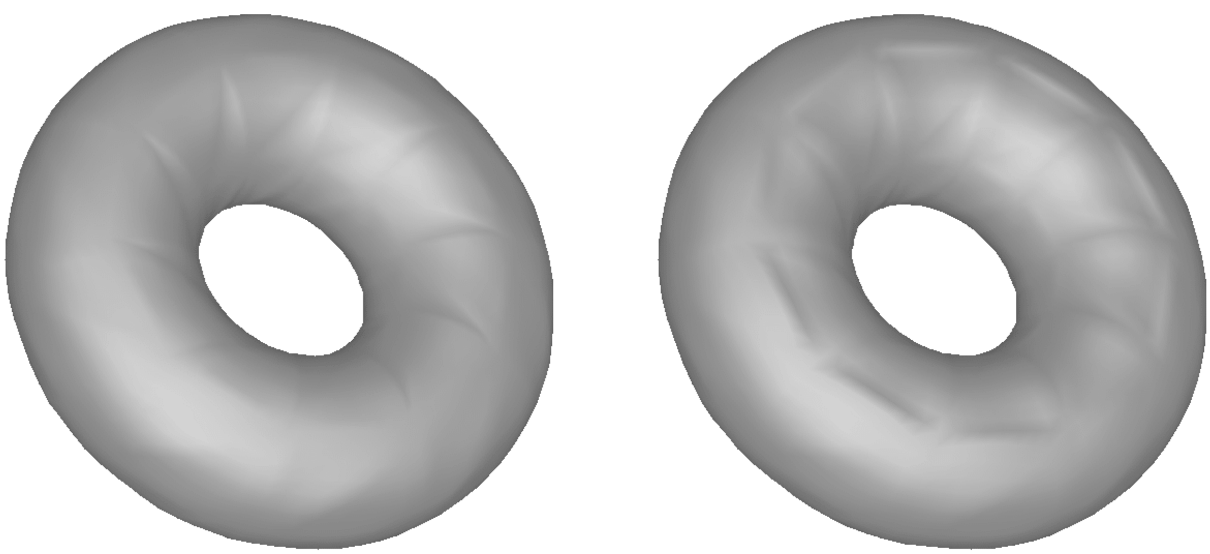

Figure 5.22: The 10 x 5 torus data set interpolated by parametric Foley-Opitz boundaries with Clough-Tocher cross boundaries, left, and parametric Foley-Opitz patches, right.

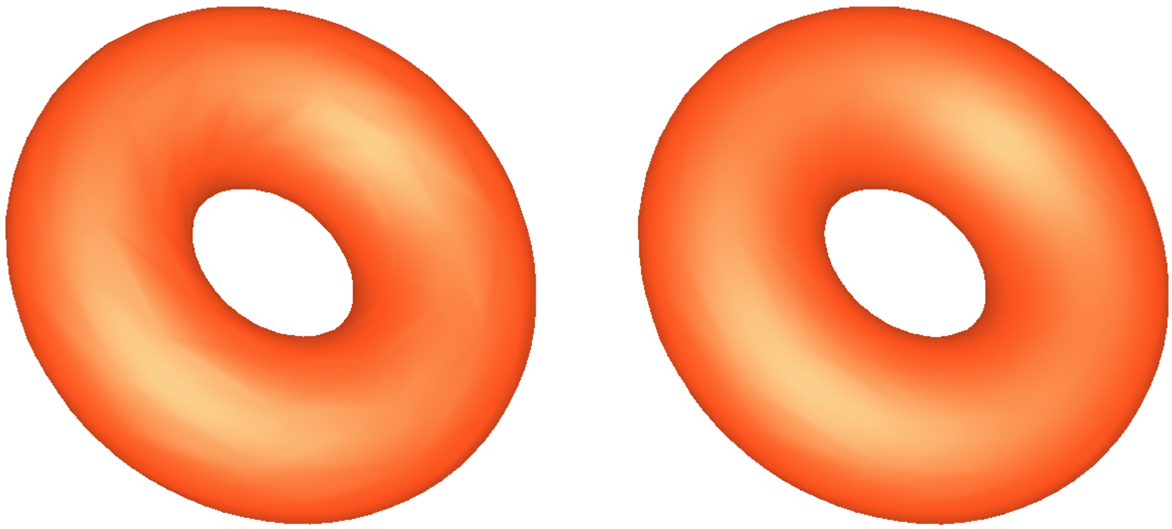

Figure 5.23: The 10 x 10 torus data set interpolated by parametric Foley-Opitz boundaries with Clough-Tocher cross boundaries, left, and parametric Foley-Opitz patches, right.

The surface with linear cross boundary derivatives is similar to the interpolants produced by Shirman-Sequin and triangular Gregory patches constructed with planar boundaries. The interpolants for the 10 x 5 torus data set, shown in Figure 5.23, again show that using linearly varying cross boundary derivatives produces surfaces similar to Shirman-Sequin and triangular Gregory patches with planar boundaries.

Since the quality of the interpolants constructed with parametric Foley-Opitz boundaries and linearly varying cross boundary derivatives are similar to the quality of Shirman-Sequin and triangular Gregory patches, the cross boundary derivatives are the deciding factor in the quality of parametric Foley-Opitz patches. The marked improvement in surface quality with the Foley-Opitz cross boundary construction compared to Clough-Tocher's cross boundary construction using the cat and the torus 10 x 10 data sets demonstrates the superiority of quadratically varying boundaries. A comparison between the torus 10 x 5 data set interpolants supports this claim since the difference in surface quality mirrors the difference discovered when comparing parametric Foley-Opitz interpolants with other schemes with this data set in Section 5.3.

5.7 Quantitative Analysis

Many data sets are produced by graphic artists or represent data taken from physical models. Quantitative comparisons with the "perfect" surface are difficult or impossible to make in these cases. When considering simple surfaces like the torus or the sphere, some quantitative analysis can be done to test the accuracy of an interpolation scheme and its performance relative to other schemes.

I have chosen two quantities to test parametric Foley-Opitz patches – the normalized distance from the model and the dot product of the interpolant's normal with the model's normal. Each of these values is measured at uniformly spaced points over the surface and the maximum and minimum values are recorded.

I have tested interpolants produced with the icosahedron, the torus 10x5, and the torus 10 x 10 data sets. Each patch is sampled uniformly at 45 positions. For parametric Foley-Opitz and triangular Gregory patches this means 800 samples for the icosahedron, 4000 samples for the the 10 x 5 torus, and 8000 samples for the 10 x 10 torus. Since Shirman-Sequin uses a split domain there will be roughly three times as many samples, thus giving results for Shirman-Sequin that are more accurate.

The following are the minimum and maximum normalized distances for parametric Foley-Opitz, triangular Gregory, and Shirman-Sequin patches, using planar boundaries for the latter two schemes. A torus would be at a constant distance of 1.

		Icosahedron	Torus 10 x 5	Torus 10 x 10
Min distance	Foley-Opitz	0.994285	0.981309	0.998741
	Triangular Gregory	0.943616	0.904166	0.974028
	Shirman-Sequin	0.976806	0.973374	0.992454
Max distance	Foley-Opitz	1.018850	1.032030	1.006930
	Triangular Gregory	1	1.004800	1.006480
	Shirman-Sequin	1	1.016100	1.008030

The icosahedron results suggest parametric Foley-Opitz patches provide a better fit since there is a smaller range of variation.

The torus 10 x 5 data set results correlate well with the images of the previous sections. The minimum distances show that triangular Gregory patches have flat areas, since the minimum is noticeably smaller than with the other two schemes. Parametric Foley-Opitz patches have a larger maximum indicating the bulges that were evident in previous images.

The torus 10 x 10 data set results also confirm the visual evidence. Parametric Foley-Opitz patches have little variation while the other two interpolants have roughly the same ranges.

The following table compares the dot products of the sampled normals with the actual normals for both the icosahedron and the tori data sets. The perfect interpolant would have dot products of 1.

	Icosahedron	Torus 10 x 5	Torus 10 x 10
Foley-Opitz	0.994756	0.944733	0.999269
Triangular Gregory	0.990964	0.969569	0.995101
Shirman-Sequin	0.998642	0.969532	0.995650

The numbers for the icosahedron indicate that parametric Foley-Opitz produces results comparable to the other two schemes, not as good as Shirman-Sequin but above the performance of triangular Gregory patches. The lower values for triangular Gregory patches indicate that some patches were flat, similar to the case of the coarsely sampled torus.

The torus 10 x 5 data shows that parametric Foley-Opitz patches perform poorly compared to the other two schemes. The sharp bulges evident in the images of the previous sections result in discrepancies in the normals.

The torus 10 x 10 data shows the improvement of parametric Foley-Opitz patches with this data set and that the other two schemes are approximately equivalent.

Chapter 6

Conclusions

The analysis of the last chapter demonstrates that parametric Foley-Opitz patches, while not perfect, provide improvements over existing local, triangular Bézier patch interpolation techniques.

In certain cases, parametric Foley-Opitz patches produced patches that were clearly unacceptable. However, I provided a simple test to decide if a patch should be rejected because of bad shape. In practice this pruning technique removed precisely those patches that did not fit the overall shape of the interpolant. Of the patches that were not pruned, there was improvement in surface quality over other techniques when fitting irregularly scattered data and finely tessellated, regularly scattered data.

In light of this improvement in surface quality, I recommend the use of parametric Foley-Opitz patches combined with another interpolation scheme. When parametric Foley-Opitz patches do not meet the tolerance selected by a surface designer, another interpolation scheme, Shirman-Sequin for instance, could be used to replace the pruned patches, giving an overall G^1 surface. Parametric Foley-Opitz patches would not be appropriate for problematic data in which there is poor correlation between the normals and the underlying data triangles.

Another positive aspect of parametric Foley-Opitz patches is the geometric nature of the shape control since the freedom in setting neighbouring patch pair continuity is characterized by a plane. In other schemes, extra degrees of freedom are left as shape parameters or are set using some heuristic. The formulation of Shirman-Sequin patches [20] demonstrates the problem with merely providing algebraic relationships between shape parameters – a set of equations restricting control point placement does not give any intuition about the shape of the patch or how modifying those equations would change the shape of a patch. On the other hand, direct manipulation of a plane with two rotational degrees of freedom, as in parametric Foley-Opitz patches, would provide a more intuitive interface.

Shape control would be the main area of future research in parametric Foley-Opitz patches. There are two facets of shape control – improving the automation of good shape construction and addressing user interface issues. Automated shape control is now handled by choosing the average of two normals to define a projection plane, however more information could be incorporated, such as other vertex normals or data triangle normals blended with various weighting functions. A good user interface would provide intuitive manipulation of individual patches or regions of a surface.

Parametric Foley-Opitz patches present a new and interesting type of surface construction primitive, most notably because of the rational blend of patch boundaries. However, they are not just of theoretical interest. These patches present a practical modeling primitive that would complement a surface construction library using other types of triangular patches.

Bibliography

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

[2] R. W. Clough and J. L. Tocher. *Finite element stiffness matrices for analysis of plate bending.* Wright-Patterson I, 1965.

[3] W. Dahmen, C.A. Micchelli and H.-P. Seidel. *Blossoming begets B-Splines built better by B-Patches.* Mathematics of Computation 59(199):97-115, July 1992.

[4] C. de Boor, K.Hollig, and M. Sabin. *High accuracy geometric Hermite interpolation.* Computer Aided Geometric Design, 4(4):269-278, December 1987.

[5] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide, Third Edition.* Academic Press, 1993.

[6] Gerald Farin. *A modified Clough-Tocher interpolant.* Computer Aided Geometric Design 19-27, 1985.

[7] Gerald Farin. *A Construction for Visual C^1 Continuity of Polynomial Surface Patches.* Computer Graphics and Image Processing 20pp 272-282 (1982).

[8] Gerald Farin and P. Kashyap. *An Iterative Clough-Tocher Interpolant.* Mathematical Modelling and Numerical Analysis Volume 26, pp 201-209, 1992.

[9] R. Farouki and J. Hinds. *On the numerical condition of polynomials in Bernstein form.* Computer Aided Geometric Design 4(3):191-216, 1987.

[10] Thomas A. Foley and Karsten Opitz. *Hybrid Cubic Bézier Triangle Patches.* Mathematical Methods in Computer Aided Geometric Design IIpp 275-286, 1992.

[11] Richard Franke. *A Critical Comparison of some Methods for Interpolation of Scattered Data.* Technical Report NPS-53-79-003, Naval Postgraduate School, Monterey, California, 1979.

[12] Richard Franke and Gregory M. Nielson. *Scattered data interpolation: A tutorial and survey.* In H. Hagen, editor, Geometric Modeling: Methods and Applications. Springer-Verlag.

[13] T. Jensen. *Assembling triangular and rectangular patches and multivariate splines.* In G. Farin, editor, Geometric Modeling: Algorithms and New Trends, pp 163-174. SIAM, 1987.

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

[15] Stephen Mann. *Using Local Optimization in Surface Fitting.* to appear in Mathematical Methods for Curves and Surfaces, 1995.

[16] Stephen Mann. *A Survey of Parametric Scattered Data Fitting Using Triangular Interpolants.* Curve and Surface Design, H. Hagen (ed), SIAM, 1992.

[17] Stephen Mann. *Cubic precision Clough-Tocher interpolation.* in progress.

[18] Tony DeRose and Stephen Mann. *An Approximately G^1 Cubic Surface Interpolant.* Mathematical Methods in Computer Aided Geometric Design II, pp 185-196, 1992.

[19] Gregory M. Nielson. *A Transfinite, Visually Continuous, Triangular Interpolant.* Geometric Modeling: Algorithms and New Trends, 1987.

[20] L. A. Shirman and C H. Séquin. *Local surface interpolation with Bézier patches.* Computer Aided Geometric Design, 4(4):279-295, December 1987.