Approximate Continuity for Functional, Triangular Bézier Patches

Yingbin Liu, Stephen Mann*

In this paper, we investigate a relaxation of the C^1 continuity conditions between functional, triangular Bézier patches, allowing for patches to meet approximately C^1. We analyze the cross boundary continuity of functional triangular Bézier patches, and derive a bound for the discontinuity in the normals between two patches based on their control points. We test our discontinuity bound on a simple data fitting scheme using cubic patches.

1 Introduction

In computer-aided geometric design, piecewise polynomial patches, such as Bézier patches, are often used to interpolate a set of data points and corresponding normal vectors. Although infinite continuity is guaranteed inside the patch, the cross boundary continuity is determined by the control points from adjacent patches. To have the required cross boundary continuity, certain rules must be applied to the control points. However, sometimes fulfillment of these conditions is impossible when using a single polynomial patch per face. As a result, more patches have to be used to achieve C^1 continuity. For example, in functional data fitting with piecewise polynomial triangular surface patches, typically three or more triangular Bézier patches are fitted to each data triangle [1, 8].

In this paper, we analyze using *approximate continuity*, instead of precise continuity, to construct triangular, functional Bézier patch surfaces, where we allow small discontinuities in the surface normals along the common boundary between two patches. Such a relaxed form of continuity should be sufficient in many applications, such as NC machining, where the machining process only approximates the desired surface. Likewise, in computer animation, if the discontinuity in surface

*School of Computer Science, University of Waterloo

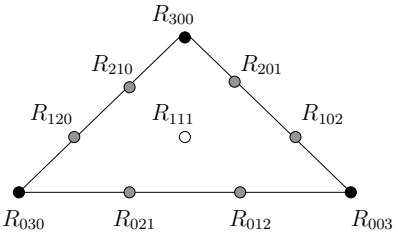


Figure 1. *Control points of a triangular Bézier patch*

normals is not visible, then an approximately continuous surface model should be sufficient, although parametric, not functional patches would be required.

Given the control points of two neighboring patches that meet with C^0 continuity, we derive two bounds on the discontinuity angle between the surface normals along the common boundary. To test the accuracy of our bound, we devised a simple data fitting scheme that fits a single cubic patch to each data triangle, where only C^0 continuity conditions are enforced.

2 Background

In this work, we are interested in triangular Bézier patch representations of functions over the plane. A triangular Bézier patch of degree n specified in barycentric coordinates relative to a domain triangle $\triangle D_0 D_1 D_2$ is given by

$$R(t) = \sum_{\vec{j},|\vec{j}|=n} R_{\vec{j}} B^n_{\vec{j}}(t),$$

where $\vec{j} = (i, j, k)$ with $i, j, k > 0$, $|\vec{j}| = i + j + k$ and

$$B^n_{\vec{j}}(t) = \frac{n!}{i!j!k!} u^i v^j w^k.$$

Here (u, v, w) are the barycentric coordinates of the parameter t relative to the domain triangle. The points $R_{\vec{j}}$ are the control points of the patch (see Figure 1). In the functional setting, we wish to represent a height field over the xy-plane, i.e., $z = f(x, y)$. If we use a Bézier patch with domain triangle $\triangle D_0 D_1 D_2$ for a function over the plane, then all of the xy-values of the control points are spaced evenly within the domain triangle, with

$$R^{xy}_{\vec{j}} = \frac{iD_0 + jD_1 + kD_2}{n},$$

where R^{xy} refers to the x, y coordinates of the three space point R (R^z will be used to refer to the z-coordinate of R). Thus, the only degrees of freedom in functional Bézier patches are the z-coordinates of the control points.

The boundaries of the triangular Bézier patch are given by the boundary control points, and the cross boundary derivative along the boundary is given by the difference of the first two layers of control points. For two Bézier patches of the same degree to meet with C^0 continuity, they must share common boundary control points. For two functional Bézier patches to meet with C^1 continuity, their adjacent panels along the boundary must be coplanar (e.g., in Figure 2, for F and $\bar{F}$ to meet with C^1 continuity, the control points F_0, H_0, $\bar{F}_0$, and H_1 must be coplanar, the control points F_1, H_1, $\bar{F}_1$, and H_2 must be coplanar, and the control points F_2, H_2, $\bar{F}_2$, and H_3 must be coplanar). For additional details on triangular Bézier patches, see [3].

For our relaxation to approximate continuity, we use the following definition [6]:

> Let S be a piecewise, C^0 surface. Define S to be ε-C^1 if the maximum angle between two surface normals at any point p on S is bounded by ε.

As stated, the definition of ε-C^1 allows for a surface to have a "razor edge"; we do not consider these surfaces to be ε-C^1 for $\varepsilon < 90$ degrees.

3 Bounding the discontinuity

In earlier work, discontinuity between adjacent triangular Bézier patches was approximated by sampling the normals of the patches along the common boundary, and computing the maximum angle between corresponding pairs of normals [2, 6]. This method has two weaknesses: first, it can be computationally expensive, and second, in general the method computes an ε that is less than the maximum. The difficulty in computing the true maximum is that the function for the discontinuity is non-polynomial. In this section, we derive a bound for the discontinuity in the surface normal based solely on the control points of the neighboring patches.

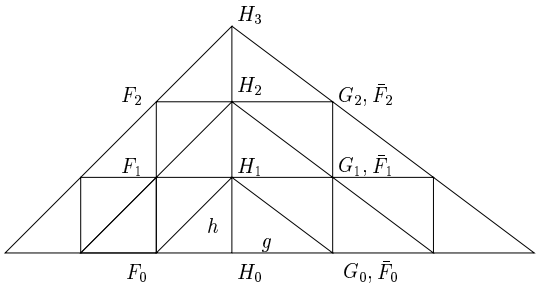


Figure 2. *Domain of patches F, $\bar{F}$, and G.*

We begin by deriving a general bound in the discontinuity between two adjacent polynomial patches. To simplify the bound, we look at the case of cubic patches that have equal normals at the ends of their common boundaries. We feel that the case of equal normals at the ends of the common boundary is interesting, since by

interpolating the position and derivatives at the corners of a data triangle, schemes can often achieve polynomial precision (cubic precision in the case of the scheme we used to generate test data for our paper). Although this polynomial precision many not provide bounds on the discontinuity, it helps to control the discontinuity between adjacent patches.

We focus on functional surfaces. Consider the case shown in Figure 2, where we have two cubic Bézier patches F and $\bar{F}$ meeting with C^1 continuity, whose common boundary we call H (this figure actually shows the orthographic projection of the control points into the xy-plane; the control points themselves have arbitrary z values). Since we are in the functional setting, the spacing between consecutive H_i^{xy} is a constant h, and the spacing between each $\bar{F}_i^{xy}$, H_i^{xy} pair is a constant g. We can compute the normal vector for patch $\bar{F}$ (which is also the normal of patch F) by calculating the cross product of the two directional derivatives in the x and y domain directions:

$$\vec{N} = \left(n \sum_{i=0}^{n-1} (H_{i+1} - H_i) B_i^{n-1}(t) \right) \times \left(n \sum_{i=0}^{n-1} (\bar{F}_i - H_i) B_i^{n-1}(t) \right).$$

To simplify the computation, we assume that the boundary H in the domain is parallel to the y-axis, and that the bottom edges of F and $\bar{F}$ in the domain are parallel to the x-axis (in Section 4, the patches can have arbitrary domain triangles; we perform a change of basis on those patches to yield the arrangement we have just described). Letting $\vec{H}_i = H_{i+1} - H_i$ and $\vec{\bar{F}}_i = \bar{F}_i - H_i$, we have $\vec{H}_i^x = 0$, $\vec{H}_i^y = h$, and $\vec{\bar{F}}_i^x = g$, $\vec{\bar{F}}_i^y = 0$. With

$$\begin{aligned} \vec{H}_i &= \vec{H}_i^x \vec{x} + \vec{H}_i^y \vec{y} + \vec{H}_i^z \vec{z} = 0\vec{x} + h\vec{y} + \vec{H}_i^z \vec{z} \\ \vec{\bar{F}}_i &= \vec{\bar{F}}_i^x \vec{x} + \vec{\bar{F}}_i^y \vec{y} + \vec{\bar{F}}_i^z \vec{z} = g\vec{x} + 0\vec{y} + \vec{\bar{F}}_i^z \vec{z} \end{aligned}$$

$\vec{N}$ can be calculated as

$$\begin{aligned} \vec{N} &= \left(n \sum_{i=0}^{n-1} [0, h, \vec{H}_i^z] B_i^{n-1}(t) \right) \times \left(n \sum_{j=0}^{n-1} [g, 0, \vec{\bar{F}}_j^z] B_j^{n-1}(t) \right) \\ &= n^2 [h \sum_{j=0}^{n-1} \vec{\bar{F}}_j^z B_j^{n-1}(t), g \sum_{i=0}^{n-1} \vec{H}_i^z B_i^{n-1}(t), -hg] \end{aligned} \tag{1}$$

Now consider a patch G over the same domain as $\bar{F}$ that shares the H_i boundary control points of F, but whose control points differ from $\bar{F}$ elsewhere. We can think of the G_i as being displaced from the $\bar{F}_i$ along the $\vec{z}$-axis: $G_j^z = \bar{F}_j^z + z_j$.

The equation for the normal $\vec{N'}$ of patch G along the boundary H is (based on Equation 1)

$$\vec{N'} = n^2 [h \sum_{j=0}^{n-1} (\vec{\bar{F}}_j^z + z_j), g \sum_{i=0}^{n-1} \vec{H}_i^z, -hg] B_i^{n-1}(t) B_j^{n-1}(t)$$

$$= \vec{N} + n^2 \sum_{j=0}^{n-1} [hz_j, 0, 0] B_j^{n-1}(t). \tag{2}$$

Letting

$$I = n^2 \sum_{j=0}^{n-1} [hz_j, 0, 0] B_j^{n-1}(t),$$

we have

$$\vec{N'} = \vec{N} + \vec{I}.$$

The relationship among $\vec{I}$, $\vec{N}$ and $\vec{N'}$ is shown as in Figure 3(a). In the following sections, we describe two methods to bound the angle between N and N'.

3.1 Method 1

The angle θ between N and N' is

$$\theta = arcsin\left(\frac{|\vec{N} \times \vec{N'}|}{|\vec{N}||\vec{N'}|}\right) = arcsin\left(\frac{|\vec{N} \times (\vec{N} + \vec{I})|}{|\vec{N}||\vec{N'}|}\right) = arcsin\left(\frac{|\vec{N} \times \vec{I}|}{|\vec{N}||\vec{N'}|}\right)$$
$$= arcsin\left(\frac{n^4 h^2 |\sum_{j=0}^{n-1} z_j B_j^{n-1}(t)[0, g, -\sum_{i=0}^{n-1} \vec{H}_i^z B_i^{n-1}(t)]|}{|\vec{N}||\vec{N'}|}\right). \tag{3}$$

Using Equation 3, it is difficult to calculate a true maximum value for θ. We can calculate an upper bound for θ by finding the maximum/minimum value for each part in the numerator and denominator separately.

Theorem 1. *For functional triangular Bézier patches F and G, arranged as shown in Figure 2 and discussed in the text, the bound on the angle θ between the normals of the two patches along their common boundaries is given by*

$$\theta \leq \arcsin\left(\frac{n^4 h^2 L_z \sqrt{g^2 + (L_h)^2}}{L_{\vec{N}} L_{\vec{N'}}}\right), \tag{4}$$

Where

$$L_z = \max\left(|\sum_{j=0}^{n-1} z_j B_j^{n-1}(t)|\right), L_h = \max\left(|\sum_{i=0}^{n-1} \vec{H}_i^z B_i^{n-1}(t)|\right),$$

and $L_{\vec{N}}/L_{\vec{N'}}$ denote the minimum length of $\vec{N}$ and $\vec{N'}$ as we vary t over $[0, 1]$.

The proof of this theorem is simple. Since L_z and L_h are degree $n-1$ Bézier curves, taking the maximum absolute value of z_j and $\vec{H}_i^z$ will generate the maximum

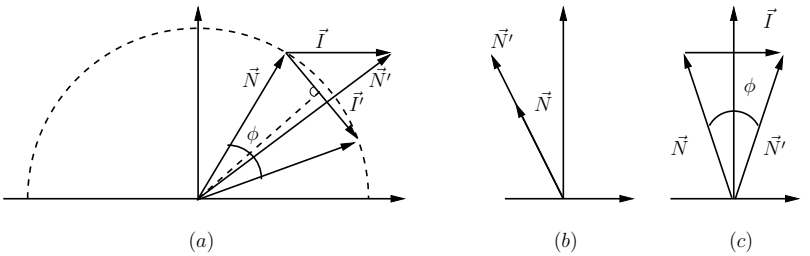


Figure 3. *Determining a bound on θ.*

values. For function $L_{\vec{N}}$, we bound

$$\begin{aligned} L_{\vec{N}} &\geq \big|n^2[h\min\big|\sum_{j=0}^{n-1}\vec{F}_j^{\,z}B_j^{n-1}(t)\big|, g\min\big|\sum_{i=0}^{n-1}\vec{H}_i^{\,z}B_i^{n-1}(t)\big|, -hg]\big| \\ &\geq n^2\big|[h\,\mathrm{Zmin}(\bar{F}_j^z), g\,\mathrm{Zmin}(\vec{H}_i^{\,z}), -hg]\big| \end{aligned}$$

where $\mathrm{Zmin}(a_i)$ is zero if the a_i are of mixed sign, and is the minimum of the absolute values of the a_i otherwise. Similiarly, we have the definition of $L_{\vec{N}'}$ as

$$L_{\vec{N}} \geq n^2\big|[h\,\mathrm{Zmin}(\bar{G}_j^z), g\,\mathrm{Zmin}(\vec{H}_i^{\,z}), -hg]\big|.$$

If patches F and G are both cubic triangular Bézier patches, and the panels at each end of the boundary are coplanar, then the control points of these end panels have zero z displacement, i.e., $G_0 = \bar{F}_0$ and $G_2 = \bar{F}_2$, and $z_0 = z_2 = 0$. Thus, the only freedom to manipulate is G_1. The upper bound defined by Equation 4 simplifies to

$$\theta \leq \arcsin\left(\frac{n^4h^2|z_1|\sqrt{g^2+(L_h)^2}}{2L_{\vec{N}}L_{\vec{N}'}}\right). \tag{5}$$

Here the extra factor of 2 in the denominator is because the Bernstein polynomial $B_j^2(t)$ obtains its maximum value over [0, 1] at $t = \frac{1}{2}$, and $B_j^2(\frac{1}{2}) = \frac{1}{2}$.

3.2 Method 2

Theorem 2. *For functional triangular Bézier patches F and G, arranged as shown in Figure 2 and discussed in the text, the bound on the angle θ between the normals of the two patches along their common boundaries is given by*

$$\theta \leq \arcsin\left(\frac{hn^2L_z}{2L}\right), \tag{6}$$

Where L_z is defined as in method 1 and

$$L = min(L_{\vec{N}}, L_{\vec{N}'}).$$

Proof. Our goal is to bound the angle θ between $\vec{N}$ and $\vec{N}'$ for $t \in [0,1]$. For now, assume that $|\vec{N}| \leq |\vec{N}'|$. As illustrated in Figure 3(a), I' is the vector that forms a chord on the circle of radius $|\vec{N}|$, where its tail is placed at the head of $\vec{N}$ with $|I| = |I'|$, and $I \cdot I' \geq 0$. Looking at Figure 3(a), we see that $\theta \leq \phi$. For any $\vec{N}$ and $\vec{I}$ of fixed length, angle ϕ reaches its maximum value when $|\vec{N}| = |\vec{N}'|$ (Figure 3(c)). We can bound θ by computing a bound on ϕ:

$$\begin{aligned}
\theta \leq \phi &\leq 2 \arcsin\left(\frac{|\vec{I}|}{2L}\right) \\
&= 2 \arcsin\left(\frac{n^2 \left|\sum_{j=0}^{n-1} [hz_j, 0, 0] B_j^{n-1}(t)\right|}{2L}\right) \\
&= 2 \arcsin\left(\frac{hn^2 L_z}{2L}\right) \qquad (7)
\end{aligned}$$

□

We now look at a special case of two cubic Bézier patches F and G, where the panels at each end of the boundary are coplanar. As mentioned in method 1, we have $G_0 = \bar{F}_0$ and $G_2 = \bar{F}_2$, therefore $z_0 = z_2 = 0$. Equation 6 simplifies to

$$\begin{aligned}
\theta &\leq 2 \arcsin\left(\frac{|h9 \sum_{j=0}^{2} z_j B_j^2(t)|}{2L}\right) \\
&\leq 2 \arcsin\left(\frac{9h|z_1|}{4L}\right). \qquad (8)
\end{aligned}$$

Again, the Bernstein polynomial $B_j^2(t)$ obtains its maximum value over [0, 1] at $t = \frac{1}{2}$, and $B_j^2(\frac{1}{2}) = \frac{1}{2}$.

For any approximate C^1 continuity surface, we can use the minimum result of the two methods as a bound of the discontinuity angle.

4 Data fitting scheme

To test our scheme, we used samplings of the Franke functions. By varying the sampling density, we obtained a set of ε-C^1 surfaces with a range of C^1 discontinuities.

The data fitting scheme solves the following problem:

Given: A triangulation $\mathcal{T}$ of a region of the plane, with z-values and normals values at the data points.

Find: A piecewise polynomial surface that interpolates the data points and the normals.

The following ε-C^1 construction is the first step of a scheme that appeared in [7]. Three types of control points are constructed: data points, boundary points and one internal point. For a triangle of data $T = \triangle P_0 P_1 P_2$, we construct a cubic Bézier patch R (whose projection in the xy-plane is shown in Figure 1) by the following steps:

1. Set the z values of the three corner data points (black points in Figure 1) to the z values of the vertices of T:

$$R^z_{3,0,0} = P^z_0, \qquad R^z_{0,3,0} = P^z_1, \qquad R^z_{0,0,3} = P^z_2.$$

   This ensures that our patch interpolates the positional data of T.

2. For the other 6 boundary control points (in gray), calculate the intersection of a line parallel to the z-axis with the tangent plane of the nearest corner point. For example, to compute $R_{2,1,0}$, we intersect the line through $R^{xy}_{2,1,0}$ in the direction of the z-axis with the plane passing through P_0 and the normal stored at P_0. The z value of this intersection point is used as the z value of $R_{2,1,0}$. The other five boundary control points are computed in a similar manner.

   This forces the patch to interpolate the normal data of T, and ensures that all the control points surrounding a data point are coplanar.

3. Set the internal control point (the white point in Figure 1) by evaluating R at the vertices of the three adjacent data triangles and averaging the result. Details of this step are given in the next section. This step is the same as the one used by Foley-Opitz to set the center point in their hybrid patch construction [4].

4.1 Setting the internal control point

For the data triangle $\triangle P_0 P_1 P_2$, and the adjacent data points Q_0, Q_1 and Q_2 (Figure 4), the last step of our construction needs to calculate the z-coordinate of the internal control point of R.

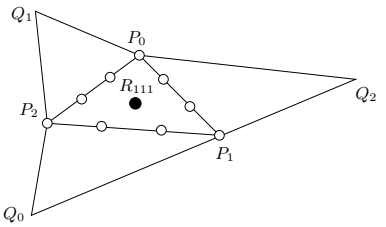


Figure 4. *Evaluation at the adjacent data points*

We will compute three values for the missing control point $R_{1,1,1}$: R_0, R_1, and R_2. Each value will be computed by evaluating the patch we are constructing, R, at

the three Q_i^{xy} values, and require $R(Q_i^{xy}) = Q_i^z$. Each evaluation gives us a linear equation in the unknown control point $R_{1,1,1}$. Solving each equation gives us one of the R_i, which we average to get the value for $R_{1,1,1}$. The details are as follows. Let (t_i^0, t_i^1, t_i^2) be the barycentric coordinates of Q_i^{xy} relative to $\triangle P_0^{xy} P_1^{xy} P_2^{xy}$. Evaluating R at Q_i^{xy} gives us

$$\begin{aligned} Q_i^z &= \sum_{\vec{\jmath},|\vec{\jmath}|=3} R_{\vec{\jmath}}^z B_{\vec{\jmath}}^3((t_i^0, t_i^1, t_i^2)) \\ &= R_i^z B_{(1,1,1)}^3(t_i^0, t_i^1, t_i^2) + \sum_{\vec{\jmath},\vec{\jmath}\neq(1,1,1)} R_{\vec{\jmath}}^z B_{\vec{\jmath}}^3((t_i^0, t_i^1, t_i^2)) \\ \Longrightarrow R_i^z &= \frac{Q_i^z - \sum_{\vec{\jmath},\vec{\jmath}\neq(1,1,1)} R_{\vec{\jmath}}^z B_{\vec{\jmath}}^3((t_i^0, t_i^1, t_i^2))}{B_{(1,1,1)}^3(t_i^0, t_i^1, t_i^2)} \end{aligned}$$

We then set $R_{1,1,1}^z = (R_0^z + R_1^z + R_2^z)/3$.

Note that if all six data points and normals (P_0, P_1, P_2, Q_0, Q_1, Q_2) come from a single cubic polynomial $\mathcal{C}$, then $R_0 = R_1 = R_2 = R_{1,1,1}$ and the surface patch we construct will reproduce this polynomial, i.e., $R = \mathcal{C}$.

While this scheme is not ideal for data fitting, it meets the requirement for generating our test data; in particular, the cubic convergence results in smaller discontinuity as we increase the sampling density of the base functions.

4.2 Boundary patches

For data fitting schemes like ours, there is usually a special construction for the boundary data triangles, since they have only one or two neighboring triangles rather than the three neighbors that internal data triangles have. For the scheme we describe in this section, the boundary triangles could be handled by, in Step 3, just computing one or two of the R_i values, and averaging those to compute $R_{1,1,1}$.

In this paper, however, our interest in the data fitting scheme is as a method for generating surface patches to test the ideas of ε-C^1 continuity. To simplify the analysis, we did not want complicating factors resulting from special cases on the boundaries. Therefore, rather than use the special case construction described in the previous paragraph, we did not construct patches for the boundary layer of triangles in the triangulation.

4.3 Applying the continuity analysis

To use the approximate continuity result of Section 3, the control points of two adjacent patches F and G must be arranged as in Figure 2 (i.e., the common boundary H must be aligned with the y-axis, and another boundary must align with the x-axis). If the domains of our patches do not meet these constraints, we can change the representation of the patches to the required configuration as follows (see Figure 5):

1. Create a coordinate frame $\mathcal{F} = \{\mathcal{O}, x', y', z'\}$, with the origin $\mathcal{O}$ at one end of the common boundary, and call the other end of the boundary as $\mathcal{E}$; align

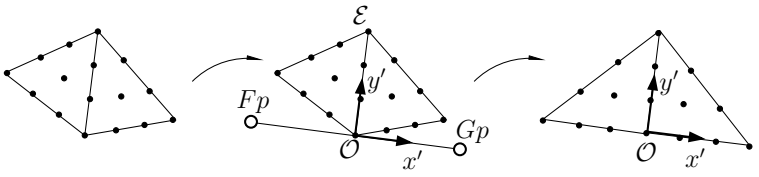


Figure 5. *Changing frames/domains for the approximate continuity calculation.*

the y' basis vector with the common boundary; set the x' basis vector to be perpendicular to y' in the xy-plane; and set the z' basis vector to be parallel to the z axis.

2. Position two points Fp and Gp in the domain along the x'-axis, with $|\mathcal{O} - Fp| = |\mathcal{O} - Gp| = |\mathcal{E} - \mathcal{O}|$.

3. Re-parameterize F by evaluating F at Fp using the de Casteljau algorithm [3], and extracting the subpatch over the domain $\triangle Fp\mathcal{O}\mathcal{E}$. A similar process can be used to extract the representation of G over the domain triangle $\triangle Gp\mathcal{E}\mathcal{O}$.

We can now use the continuity bounds described in Section 3, using the coordinates of the control points relative to $\mathcal{F}$. To use the result in Section 3, we have to know the position of the control points that have precise continuity with F (e.g., the $\bar{F}_i$ of Figure 2). The points $\bar{F}_i$ will have the same xy-positions as the G_i, but have a z-value so that patch $\bar{F}$ meets patch F with C^1 continuity. The points $\bar{F}_0$ and $\bar{F}_2$ already meet the required C^1 planarity conditions; we just need to find $\bar{F}_1$, which is given by

$$\bar{F}_1 = t_0 H_1 + t_1 H_2 + t_3 F_1,$$

where (t_0, t_1, t_2) are the barycentric coordinates of G_1^{xy} relative to $\triangle H_1^{xy} H_2^{xy} F_1^{xy}$.

5 Tests and results

For our tests, we sampled the six "Franke functions" as described by Grandine [5], and fit approximate C^1 surfaces to this data using the scheme described in Section 4. Since we wished to test the boundary discontinuity and the convergence properties as we increased the sampling, we used uniform samplings of these six functions.

We bounded the discontinuity in the surface normals between adjacent patches using three methods. The first was by sampling the common boundary of every pair of adjacent patches at 100 points, computing the surface normals, and calculating the angle between these normals. The other two methods are using the bounding methods we describe in Section 3 of this paper. Table 1 shows the maximum discontinuity of the sampled angles and correspondent two upper bounds for all curves; all the numbers are in degree.

The Grid column refers to the number of cells in the square grid to which we fit patches; two triangular patches were fit to each square of the grid. We did

Table 1. *Discontinuity of Franke function surfaces.*

Functions	Grid	Discontinuity	Method 1	Method 2
F1	10×10	26.487492	67.894637	35.814840
	20×20	8.254011	11.494557	8.597684
	40×40	1.226258	1.469088	1.266205
F2	10×10	8.059139	13.315756	11.347997
	20×20	2.676044	3.660120	3.431178
	40×40	0.438722	0.515085	0.541983
F3	10×10	1.430003	1.630511	1.498616
	20×20	0.231934	0.240315	0.235121
	40×40	0.030659	0.030940	0.030764
F4	10×10	0.512760	0.545877	0.519773
	20×20	0.069462	0.070669	0.069708
	40×40	0.008862	0.008901	0.008870
F5	10×10	5.426008	9.497992	6.355084
	20×20	0.988369	1.235251	1.039557
	40×40	0.137503	0.146953	0.139415
F6	10×10	0.432433	0.503140	0.506612
	20×20	0.081190	0.089400	0.097062
	40×40	0.013420	0.014184	0.016166

not fit data to the boundary layer of squares, so the actual number of samples we took from the corresponding Franke function was three higher in each dimension (e.g., a 10×10 grid means we took 13×13 samples). The samples were taken so that (ignoring the boundary layer) the outer layer aligns with the $[0, 1]$-square in the domain. As shown in Table 1, although the discontinuity of our scheme is noticeable at low samplings of the Franke function, the discontinuity drops quickly when sampling density is increased; the discontinuity appears to drop as $O(h)$; however, we have not proven this result.

From Table 1, our bounds on the discontinuity appear to be a tight bound on the actual discontinuity. The bound is not quite satisfying for some surfaces such as method 1 for function 1 at grid 10×10, but it improves as the sample density is increased. Comparing the two bounds, we find that bound of method 1 is not as good as the bound for method 2 for most surfaces. If we look at a plot of the normal discontinuities for all the patch boundaries (Figure 6(d)), we again see that the maximum discontinuity for most boundaries is small, and as expected that the maximum for any one boundary occurs near the middle of the curve but not exactly at $t = 0.5$.

If we plot the ratio of the estimated discontinuity to the maximum sampled discontinuity for all boundaries of one surface, we see that the ratio can be higher than the value in Table 1. Plots were generated for Function 1, on a 20×20 grid (Figure 6(a,b,c)). Each point on the graph is the ratio of our error bound to the numerically computed maximum error for the boundary between two pairs of patches. From the plot, we see that using method 1 (Figure 6(a)) can generate more boundary curves with lower ratio than using method 2 (Figure 6(b)), especially for

Table 2. *Ratio distribution for Franke Function 1.*

Ratio	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-
Method 1	84.43	12.76	1.98	0.34	0.09	0.0
Method 2	77.33	15.86	4.57	1.29	0.60	0.34
Combined	94.83	5.17	0.0	0.0	0.0	0.0

Table 3. *Ratio distribution for other Franke functions.*

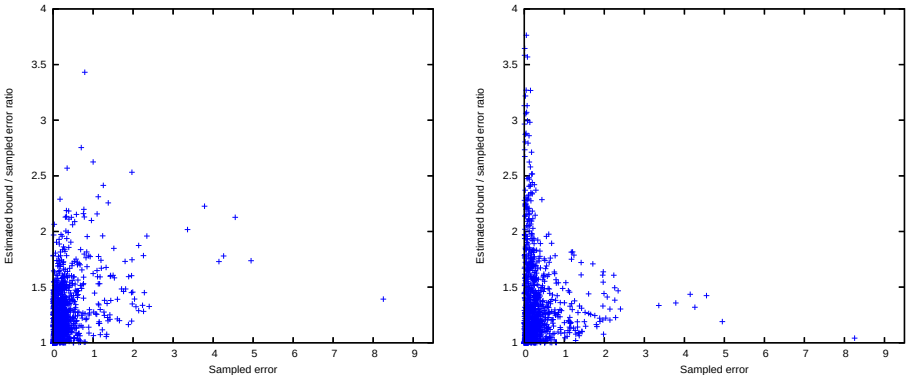
	Ratio	1.0-1.1	1.1-1.2	1.2-1.3	1.3-1.4	1.4-1.5	1.5-
F2	Method 1	71.72	5.86	9.48	6.21	0.00	6.72
	Method 2	74.66	8.97	9.48	6.55	0.00	0.34
	Combined	74.66	8.97	16.03	0.0	0.00	0.34
F3	Method 1	90.17	9.57	0.26	0.00	0.00	0.00
	Method 2	71.43	17.84	8.71	3.02	0.00	0.00
	Combined	96.12	3.88	0.00	0.0	0.00	0.00
F4	Method 1	100.0	0.00	0.00	0.00	0.00	0.00
	Method 2	68.10	31.90	0.00	0.00	0.00	0.00
	Combined	100.0	0.0	0.00	0.00	0.00	0.00
F5	Method 1	68.97	18.97	7.41	3.28	1.03	0.34
	Method 2	72.24	10.34	6.90	4.48	1.90	4.14
	Combined	78.45	15.34	4.48	1.38	0.34	0.0
F6	Method 1	96.03	3.79	0.17	0.00	0.00	0.00
	Method 2	75.00	21.38	2.24	0.86	0.34	0.17
	Combined	99.31	0.69	0.0	0.0	0.0	0.0

those with lower discontinuity angles. But method 2 is better for boundary curves where there is a high normal discontinuity. We can take the minimum of the two results as the final bound (Figure 6(c)). Table 2 shows the percentage of boundaries for which the ratio of our error bound methods to the sampled method lie within different ranges for function 1. Table 3 shows the percentage distribution for other functions. All the surfaces in Table 2 and Table 3 are sampled on a 20×20 grid. For the combination of the two methods, the majority of the points are in the lower end of the ratio (Figure 6(c)).

6 Future work

In this paper, we focused on the aproximate continuity of cubic Bézier patches in the functional case. This work is really a first step in working with approximate continuity; although we derived an upper bound for the cross boundary discontinuity, we would like to compute a tighter bound, and extend the bound to arbitrary degree. Alternatively, rather than compute the bound based on the control points of the patch, we might be able to base a discontinuity bound on samples of the normals along the patches.

The construction we gave here built patches and then tested the normal discontinuity. The data fitting scheme in this paper is only for testing purpose, but we



(a) Bound over sampled error ratio, method 1. (b) Bound over sampled error ratio, method 2.

(c) Bound over sampled error ratio, minimum. (d) Discontinuity curves.

Figure 6. *Plots of Franke function 1.*

are also interested in making a construction that guarantees a level of approximate continuity, one that perhaps trades off the level of discontinuity with curvature properties of the patch.

Additionally, we want to extend the approximate continuity construction to parametric surfaces. This will be more complex, as many of the simplifications we used in computing the bound for the functional case do not apply in the parametric setting.

As a final note, it is important to realize that the visibility of the discontinuity depends not just on the angle between the normals, but also on the material properties, the viewing angle, etc. Perceptual studies should be made to determine the maximum angle of allowable discontinuity for various material properties.

Bibliography

[1] R Clough and J Tocher. Finite element stiffness matrices for analysis of plates in bending. In *Proceedings of Conference on Matrix Methods in Structural Analysis*, 1965.

[2] Tony DeRose and Stephen Mann. An approximately G^1 surface interpolant. In T. Lyche and L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design II*, pages 185–196. Academic Press, 1992.

[3] Gerald Farin. *Curves and Surfaces for CAGD: A Practical Guide*. Academic Press, fifth edition, 2002.

[4] Thomas A. Foley and Karsten Opitz. Hybrid cubic Bézier triangle patches. In T Lyche and L Schumaker, editors, *Mathematical Methods for Computer Aided Geometric Design II*, pages 275–286. Academic Press, 1992.

[5] Thomas Grandine. An iterative method for computing multivariate C^1 piecewise polynomial interpolants. *Computer Aided Geometric Design*, 4:307–319, 1987.

[6] Stephen Mann. *Surface Approximation Using Geometric Hermite Patches*. PhD thesis, University of Washington, 1992.

[7] Stephen Mann. Implementation of some triangular data fitting schemes using averaging to get continuity. Technical Report Research Report, Computer Science Department, University of Waterloo, April 2000.

[8] Michael Powell and Malcolm Sabin. Piecewise quadratic approximations on triangles. *ACM Transactions of Mathematical Software*, 3(4):316–325, December 1977.