Cover Page

Title: Surface Approximation Using Geometric Hermite Patches

Abstract:

A high-order-of-approximation surface patch is used to construct continuous, approximating surfaces. This patch, together with a relaxation of tangent plane continuity, is used to approximate offset surfaces, algebraic surfaces, and S-patches.

Keywords: Triangular Bézier patches, approximate continuity

Name: Stephen Mann
Affiliation: University of Waterloo
Email address: smann@cgl.uwaterloo.ca
Postal address:

Computer Science Department
University of Waterloo
200 University Ave W
Waterloo, Ontario N2L 3G1 CANADA

Phone: +1 519 888 4567 x4526
FAX: +1 519 885 1208

Surface Approximation Using Geometric Hermite Patches

Stephen Mann
Computer Science Department
University of Waterloo
200 University Ave W
Waterloo, Ontario N2L 3G1
Canada
smann@cgl.uwaterloo.ca

Abstract

A high-order-of-approximation surface patch is used to construct continuous, approximating surfaces. This patch, together with a relaxation of tangent plane continuity, is used to approximate offset surfaces, algebraic surfaces, and S-patches.

1. Introduction

Many researchers have investigated the problem of interpolating a triangulated set of data. For parametric data, a variety of approaches have been proposed, most of which create G^1 surfaces that interpolate position and normal information at the vertices. Unfortunately, while these schemes produce surfaces that are mathematically smooth, the surfaces usually fail to look smooth. (See [LMD92, MLL$^+$92] for surveys of triangular, parametric interpolation schemes, and for a discussion of problems with these schemes.)

One problem with the G^1 schemes is that a fairly high degree surface patch is required to satisfy the continuity conditions. However, after meeting the continuity constraints, several degrees of freedom remain unset. While these degrees of freedom can be used as shape parameters, they should be given good default values automatically. Typically, these degrees of freedom are set using simple heuristics. Unfortunately, these simple choices yield surfaces with severe shape defects.

There are several approaches to improving the settings of these excess degrees of freedom. Several researchers are investigating using global optimization for setting them [CG91, MS92]. The resulting surfaces, while computationally expensive to construct, have better shape than those constructed using simple heuristics.

Subdivision surfaces provide an alternative surface construction approach. Many subdivision schemes (such as Catmull-Clark and Loop subdivision [CC78, Loo87]) are approximating schemes and not well suited to this problem. Others, such as the butterfly scheme [DLG90] are interpolatory, and could be used to solve this problem. See, for example, Warren's course notes for an introduction to subdivision schemes [War98].

A different approach is taken in this paper. Rather than developing improved settings of the excess degrees of freedom, I use a cubic patch technique of DeRose-Mann that leaves few degrees of freedom unset after matching geometric constraints. The resulting surface patch has better shape the those built by the G^1 schemes. Then, to construct a piecewise polynomial surface, I relax the continuity conditions from G^1 to ε-G^1 (ε-G^1 is defined in Section 3.1).

In the next section, I review the cubic interpolant, a patch that interpolates the data at the corners of a triangle to second order. Then in Section 3, I introduce the notion of approximate continuity (ε-G^1) and show how to use the cubic interpolant to produce ε-G^1 approximations to offset surfaces, algebraic surfaces, and S-patches.

2. The Cubic Interpolant

There are several free parameters in most G^1 schemes. Experience has shown that using simple heuristics to set these parameters leads to surfaces that fail to appear visually smooth. To improve surface quality, one could use more principled settings of the remaining degrees of freedom, using for example, numerical optimization. Alternatively, for surface approximation, the excess degrees of freedom could be set using additional data sampled from the underlying surface. In this section, I take the latter approach and review a construction of DeRose-Mann that sets the degrees of freedom in a degree three triangular Bézier patch by matching higher order derivative information at the data

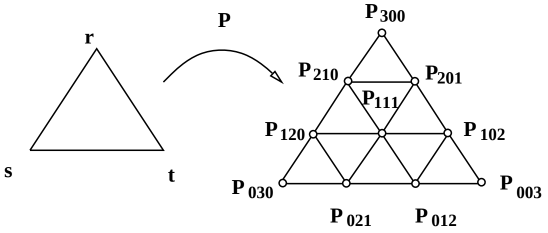

Figure 1. The domain triangle and control points of a cubic Bézier patch $\mathbf{P}$

points.

If the data given at the vertices includes position, normal, and curvature, then in general, we can interpolate it with triangular, cubic Bézier patches. In the following, a single patch will be called $\mathbf{P}$. The domain of $\mathbf{P}$ is given by the triangle $\triangle\mathbf{rst}$. The control points of $\mathbf{P}$ are denoted with standard multi-index notation (Figure 1). The problem we wish to solve is:

> Given: Three points, with associated normals and second fundamental forms, $(\mathbf{V}_i,\ \widehat{\mathbf{N}}_i,\ \mathrm{I\!I}_i)$, $i \in \{\mathbf{r}, \mathbf{s}, \mathbf{t}\}$.
>
> Find: A cubic polynomial surface patch that interpolates this data.

The data at each input point imposes eight constraints on the patch: three for position, two for the normal, and three for the second fundamental form. For two linearly independent vectors $\vec{\mathbf{u}}$ and $\vec{\mathbf{v}}$ in the tangent plane of a patch $\mathbf{P}$ at $\mathbf{V}_r$, $\mathbf{P}$ must satisfy the following constraints to interpolate the data at $\mathbf{V}_r$:

$$
\begin{aligned}
\mathbf{P}(\mathbf{r}) &= \mathbf{V}_r, \\
< \mathbf{D}_{\vec{\mathbf{u}}}\mathbf{P}(\mathbf{r}),\ \widehat{\mathbf{N}}_r > &= 0, \\
< \mathbf{D}_{\vec{\mathbf{v}}}\mathbf{P}(\mathbf{r}),\ \widehat{\mathbf{N}}_r > &= 0, \\
< \mathbf{D}_{\vec{\mathbf{u}}\vec{\mathbf{u}}}\mathbf{P}(\mathbf{r}),\ \widehat{\mathbf{N}}_r > &= -\mathrm{I\!I}_r(\vec{\mathbf{u}}, \vec{\mathbf{u}}), \\
< \mathbf{D}_{\vec{\mathbf{v}}\vec{\mathbf{v}}}\mathbf{P}(\mathbf{r}),\ \widehat{\mathbf{N}}_r > &= -\mathrm{I\!I}_r(\vec{\mathbf{v}}, \vec{\mathbf{v}}), \\
< \mathbf{D}_{\vec{\mathbf{u}}\vec{\mathbf{v}}}\mathbf{P}(\mathbf{r}),\ \widehat{\mathbf{N}}_r > &= -\mathrm{I\!I}_r(\vec{\mathbf{u}}, \vec{\mathbf{v}}).
\end{aligned}
$$

Similar constraints are imposed by the other two corners. The total number of constraints on the patch is therefore 24. A quadratic polynomial patch has only 18 degrees of freedom and cannot in general be used to solve this problem. A cubic patch is the minimum degree polynomial patch that can be used; it has 30 degrees of freedom in its ten control points.

The construction proceeds as follows: each boundary of the patch is built to interpolate the data that two corners with a planar geometric Hermite curve using the method of de Boor-Hollig-Sabin [dBHS87]. This leaves unset the center control point (P_{111}). Each second fundamental forms at the data points place a planar constraint on this control point; we find its setting by intersecting these three planes as described in the next paragraph.

The center control point influences the mixed partial at the three corners of the patch. Letting $\vec{\mathbf{u}} = 3(\mathbf{P}_{210} - \mathbf{P}_{300})$ and $\vec{\mathbf{v}} = 3(\mathbf{P}_{201} - \mathbf{P}_{300})$, the mixed partial's normal component at the corner $\mathbf{V}_r$ is

$$
\begin{aligned}
\mathrm{I\!I}_r(\vec{\mathbf{u}}, \vec{\mathbf{v}}) &= - < \mathbf{D}_{\vec{\mathbf{u}}\vec{\mathbf{v}}}\mathbf{P}(r),\ \widehat{\mathbf{N}}_r > \\
&= - < 6(\mathbf{P}_{111} + \mathbf{P}_{300} - \mathbf{P}_{210} - \mathbf{P}_{201}),\ \widehat{\mathbf{N}}_r > \\
&= -6 < \mathbf{P}_{111} - \mathbf{P}_{300},\ \widehat{\mathbf{N}}_r > .
\end{aligned}
$$

Thus, $\mathbf{P}$ agrees with the mixed partial information at $\mathbf{V}_r$ if $\mathbf{P}_{111}$ is placed anywhere in the plane passing though the point $\mathbf{P}_{300} - \frac{\mathrm{I\!I}_r(\vec{\mathbf{u}},\vec{\mathbf{v}})}{6}\widehat{\mathbf{N}}_r$ and perpendicular to $\widehat{\mathbf{N}}_r$. Each corner restricts $\mathbf{P}_{111}$ to such a plane. In general, these three planes can be intersected to find a unique position for $\mathbf{P}_{111}$ that satisfies all the input constraints.

Details of this construction can be found in a paper by DeRose-Mann [DM92]. I shall refer to this patch as the *cubic interpolant*. It is conjectured that this patch has fifth order approximation [Man92].

3. Applications

Most interpolation techniques construct a piecewise continuous surface with patches that meet with tangent plane continuity. Many applications, however, build a physical approximation to the mathematical surface that is not tangent plane continuous. For example, in a manufacturing process, machining tools create surfaces that are only approximately continuous. Thus, it may be advantageous to relax the continuity conditions earlier in the design process if it simplifies the construction, simplifies the representation, improves efficiency, or improves the surface shape. In this section, I use the cubic interpolant to construct approximately tangent plane continuous surfaces.

3.1 Approximate Continuity

In the previous section, a surface patch was fit to a triangle of data without considering the discontinuity in surface normals between neighboring patches. As all the degrees of freedom in each patch are used to match derivative information, we are unable to form a G^1 join in general. To ensure some degree of smoothness, I use the following relaxation of tangent plane continuity:

> **Definition:** Let $\mathbf{S}$ be a piecewise C^1, globally C^0 surface. Define $\mathbf{S}$ to be ε-G^1 if the the maximum

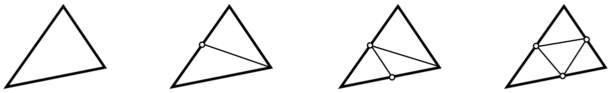

Figure 2. Mesh refinement.

angle between two surface normals at any point $\mathbf{p}$ on $\mathbf{S}$ is bounded by ε.

As stated, the definition of ε-G^1 allow for a surface to have a "razor edge". However, such surfaces should not be considered ε-G^1 for $\varepsilon \leq 90$ degrees. Note that a G^1 surface is ε-G^1 for every $\varepsilon \geq 0$.

3.2 A Piecewise Cubic Surface Scheme

In this section, I show how to create an ε-G^1 surface with the cubic interpolant that interpolates a triangular net of data. For a triangular net of data, we can construct a cubic interpolant patch for each face. A surface constructed in this way is globally C^0, and G^2 at the data points (although occasionally the second order data can not be interpolated, in which case the surface is only G^1 at those vertices).

However, neighboring patches of this surface may join with large G^1 discontinuities along boundary curves. So after constructing the surface, the discontinuities between normals on the boundaries are checked. If the discontinuity along an edge is determined to be small enough, the edge is left unchanged. If the discontinuity is large, then the mesh is refined adaptively in that region as shown in Figure 2.

The refinement process generates new data points. At each new point, the position, normal and surface curvature must be determined. When approximating a known surface, this data is sampled from the surface. The cubic interpolant is then used to construct patches for the new faces, and the process is repeated. If the conjecture mentioned at the end of Section 2 holds, then we expect this process to terminate quickly.

One slight difficulty occurs when the cubic interpolant is used to approximate a surface with a smooth boundary. The above method restricts the boundaries of the cubic patches to be planar. Since the particular plane used for each patch boundary is chosen independent of the other patches, the resulting boundary of the surface will not in general be G^1. To achieve a G^1 surface boundary, I sample the tangents of the surface along the boundary and fit a non-planar geometric Hermite space curve to the data [Man92].

To ensure that the piecewise cubic surface is ε-G^1 for a given ε, we need to bound the discontinuity in surface normals between two cubic patches. Currently, I sample the normals of both patches at ten locations on their common boundary and use the maximum discontinuity between these pairs of normals as an estimate of the maximum discontinuity along the entire boundary.

4. Approximation of Known Functions

One application of the cubic interpolant is to approximate known functions. In this section, I use the cubic interpolant to approximate algebraic surfaces, offset surfaces, and S-patches. For all three applications, the primary problem is the computation of the second fundamental forms.

The question of "how large a discontinuity in the normal is acceptable" is a difficult question to answer. For visual smoothness, the allowable deviation from equal normals will depend on the material properties of the surface. I will evaluate surface quality by visually examining the surface. For the red, diffuse material used in this paper, I empirically determined that angles less than one degree are not visible in shaded images, and used one degree as my stopping criteria for the subdivision. Note that the visibility of the discontinuity is also a function of the viewing angle and of the direction to the light source. In addition to using a bound on the angle, the surfaces shown in this paper were evaluated by rotating the objects and looking at them from various angles.

4.1 Algebraic surfaces

An algebraic surface is an implicit polynomial surface. That is, it is the set of all points such that the equation

$$P(x, y, z) = 0$$

is satisfied, where P is a polynomial in x, y, and z. Algebraic surfaces and parametric polynomial surfaces both have uses in CAGD. Sometimes it is useful to convert between the two representations. While a parametric surface can always be converted to implicit form, it is sometimes impossible to convert an algebraic surface to a parametric polynomial surface. If an approximation to the algebraic surface is acceptable, then the cubic interpolant can be used to obtain a piecewise polynomial approximation.

Since the evaluation of an algebraic function is zero for points on the surface, as a first step we must locate points on the implicit surface. This is typically done by searching a portion of space, leading to the related difficulty of determining which region of space to search. Here, I assume that the region of interest is given, and I use a variation of marching cubes to find an initial sampling of the surface [HDD$^+$92].

The data points in the marching cubes mesh are only near the surface. Thus, the points are projected back onto the implicit surface, and normals and curvature data are added using the method of Schweitzer and DeRose [SD]. Schweitzer and DeRose show that if f is the blossom [Ram87] of an algebraic function F, and $\mathbf{p}$ is a point on F, then the normal of F at $\mathbf{p}$ is given by

$$\widehat{\mathbf{N}}(\mathbf{p}) = \frac{\sum_{i=1}^{3} f(\hat{\mathbf{e}}_i, \mathbf{p}, \ldots \mathbf{p})\hat{\mathbf{e}}_i}{(\sum_{i=1}^{3} f(\hat{\mathbf{e}}_i, \mathbf{p}, \ldots \mathbf{p})^2)^{1/2}},$$

where $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3\}$ is a basis for the space. The second fundamental form of $\mathbf{F}(\mathbf{p}) = 0$ is

$$\mathrm{II}_{\mathbf{p}}(\vec{\mathbf{u}}, \vec{\mathbf{v}}) = \frac{(n-1)f(\vec{\mathbf{u}}, \vec{\mathbf{v}}, \mathbf{p}, \ldots, \mathbf{p})}{(\sum_{i=1}^{3} f(\hat{\mathbf{e}}_i, \mathbf{p}, \ldots \mathbf{p})^2)^{1/2}}.$$

Starting from this initial mesh, the cubic interpolant is fit to the data, and the mesh is refined where needed.

Meshes for the function $4x^2+2y^2+z^2-4=0$ are given in Figure 3. The number of faces in each mesh appears below the mesh. The left most mesh is the initial mesh, and the remaining four show the mesh after 1, 2, 3, and 4 levels of refinement. In Figure 7, the Gaussian curvature plots of the surface corresponding to these meshes are shown in two views: one matching the view of the meshes in Figure 3, and one from the top. The image in the bottom right is a shaded image of the final surface. The final surface is ε-G^1 for $\varepsilon = 0.4$ degrees.

The above example is fairly simple. In particular, there are no degeneracies in the surface (i.e., singular points, singular lines, etc). In the presence of degeneracies, a different algorithm would have to be devised to compute the initial mesh, as the marching cubes algorithm fails to account for the degeneracies.

4.2 Offset surfaces

Given a surface $\mathbf{S}$, the offset surface $\mathbf{S}^*$ is computed by moving a fixed distance along the normals to $\mathbf{S}$. More precisely, if $\widehat{\mathbf{N}}_{\mathbf{S}}$ is the normal to $\mathbf{S}$, then

$$\mathbf{S}^*(\mathbf{t}) = \mathbf{S}(\mathbf{t}) + d\widehat{\mathbf{N}}_{\mathbf{S}}(\mathbf{t}),$$

where d is a fixed constant. Offset surfaces have applications in NC-machining, where the point of tool motion is offset from the surfacing being machined.

An offset surface is typically more complex mathematically than the base surface. For example, if $\mathbf{S}$ is a parametric polynomial surface, then the offset to $\mathbf{S}$ is not in general a parametric polynomial surface. This causes difficulties for many modeling systems, as such systems are equipped to deal with parametric polynomial surfaces but cannot explicitly represent offsets to polynomial surfaces. Often, however, users are satisfied with an approximation to the offset surface.

To approximate offsets to surfaces with the cubic interpolant, the offset surface must be sampled at a variety of locations for position, tangent plane, and second fundamental form. If we have a sampling of the base surface, then a sampling of the offset surface can be computed as follows: The position is given by the definition of the offset surface. The mapping of a vector $\vec{\mathbf{v}}$ in the tangent plane of $\mathbf{S}$ to $\vec{\mathbf{v}}^*$ in the tangent plane of $\mathbf{S}^*$ is given by the following:

Refinement Level	Color
6+	white
5	light green
4	lavender
3	red
2	blue
1	yellow-brown
0	dark green

Table 1. Refinement color map.

$$\begin{aligned} \mathbf{D}_{\vec{\mathbf{v}}^*}\mathbf{S}^* &= \mathbf{D}_{\vec{\mathbf{v}}}\mathbf{S} + d\mathbf{D}_{\vec{\mathbf{v}}}\widehat{\mathbf{N}}_{\mathbf{S}} \\ &= \vec{\mathbf{v}} - d(\mathrm{II}^{\mathbf{S}}(\vec{\mathbf{v}}, \hat{\mathbf{p}}_1)\hat{\mathbf{p}}_1 + \mathrm{II}^{\mathbf{S}}(\vec{\mathbf{v}}, \hat{\mathbf{p}}_2)\hat{\mathbf{p}}_2), \quad (1) \end{aligned}$$

where $\hat{\mathbf{p}}_1$ and $\hat{\mathbf{p}}_2$ are the principal directions of $\mathbf{S}$. Note that $\vec{\mathbf{v}}^*$ is a linear combination of vectors in the tangent plane of $\mathbf{S}$. Thus, we have

$$\widehat{\mathbf{N}}_{\mathbf{S}^*}(\mathbf{p}) = \widehat{\mathbf{N}}_{\mathbf{S}}(\mathbf{p}).$$

Further note that $\vec{\mathbf{v}}*$ is parallel to $\vec{\mathbf{v}}$ if and only if $\vec{\mathbf{v}}$ is a principal direction. This fact may be used to show that principle directions on the base surface map to principal directions on the offset surface.

The mapping of curvatures in a principal direction $\vec{\mathbf{p}}$ having curvature k is

$$\begin{aligned} \vec{\mathbf{p}}* &= \vec{\mathbf{p}} - d\mathrm{II}(\vec{\mathbf{p}}, \vec{\mathbf{p}})\vec{\mathbf{p}} \\ &= (1-dk)\vec{\mathbf{p}}, \end{aligned}$$

implying that the curvature of the offset surface in direction $\vec{\mathbf{p}}*$ is $k/(1-dk)$. The mapping of the principal directions gives us a complete characterization of the second fundamental form of $\mathbf{S}^*$.

On the left in Figure 8 is an offset to a bicubic tensor product B-spline surface consisting of nine bicubic patches. The cubic interpolant approximation to this surface appears in the center of this figure. The cubic interpolant surface was refined seven times; the surface consists of 148 patches and is ε-G^1 for $\varepsilon = 0.72$ degrees. A map showing the refinement level appears on the right in the figure. Table 1 gives the mapping of colors to levels of refinement. The small region that was refined six and seven times occurs along the surface's right edge.

Note that in this example, the boundary of the offset surface is composed of four G^1 curves. To construct G^1 boundaries for the approximation, I sampled the tangents of the boundary of the offset surface (Equation 1) and constructed non-planar geometric Hermite curves.

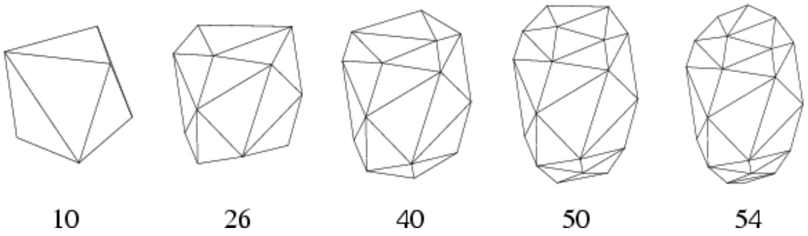

Figure 3. Meshes for the function $4x^2 + 2y^2 + z^2 - 4 = 0$.

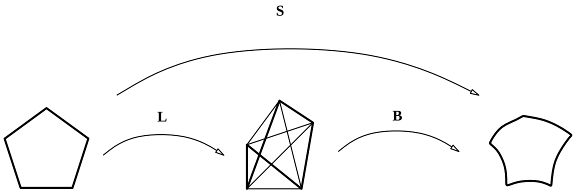

Figure 4. A five sided S-patch.

4.3 S-Patches

For most purposes in CAGD, we can use three and four sided surface patches. Sometimes, however, n-sided patches are needed, where n is greater than four. An S-patch is one type of patch that can interpolate a face with an arbitrary number of vertices. However, there are several difficulties in using S-patches, two of which are of interest to us here. First, S-patches have high rational degree, making them computationally expensive to evaluate. And second, the representation of S-patches with many sides is large. In this section I show how to approximate a G^1 S-patch network with the cubic interpolant, reducing both the computation and storage costs.

An S-patch is the composition of two maps, $\mathbf{S} = \mathbf{B} \circ \mathbf{L}$. The map $\mathbf{L}$ is a rational map from a regular n-gon to an $n-1$-simplex; the map $\mathbf{B}$ is a Bézier simplex. Thus, the composition $\mathbf{S}$ is a map from an n-gon to an n-sided surface patch (Figure 4). The *depth* of an S-patch is defined to be the degree of $\mathbf{B}$. Loop and DeRose give a detailed description of S-patches [LD89].

A few properties of S-patches are worth mentioning here. First, S-patches are a generalization of both triangular Bézier patches and of tensor product Bézier patches: three sided S-patches are Bézier patches, and tensor product Bézier patches are a special case of four sided S-patches. A second property is that while in general S-patches are rational polynomial surfaces, their boundaries are polynomial curves.

The first step in creating an ε-G^1 approximation to an S-patch network is to find an initial triangulation of the S-patches. The easiest approach is to subdivide each S-patch at the center of its domain and split the n-sided patch into n triangular regions. Each one of these triangular regions can now be approximated with a cubic interpolant patch.

To make such an approximation, we must compute normals and second fundamental forms at the sample points. Since we know the rational polynomial equation for the S-patch, we can compute these values directly. Essentially, $\mathbf{L}$ is a function of two variables, say x and y, and $\mathbf{B}$ is a Bézier simplex. Blossoming $\mathbf{B}$ gives the following form for $\mathbf{S}$:

$$\begin{aligned}\mathbf{S}(x,y) &= \mathbf{B}(\mathbf{L}(x,y)) \\ &= \mathbf{b}(\mathbf{L}(x,y),\ldots,\mathbf{L}(x,y)).\end{aligned}$$

To compute the first and second fundamental form, we need to take partial derivatives of $\mathbf{S}$:

$$\begin{aligned}\mathbf{D}_x\mathbf{S}(x,y) &= d\mathbf{b}\left(\mathbf{D}_x\mathbf{L}(x,y),\ldots,\mathbf{L}(x,y)\right), \\ \mathbf{D}_{x,x}\mathbf{S}(x,y) &= d(d-1)\mathbf{b}\Big(\mathbf{D}_x\mathbf{L}(x,y),\mathbf{D}_x\mathbf{L}(x,y), \\ &\qquad \mathbf{L}(x,y),\ldots,\mathbf{L}(x,y)\Big) + \\ &\quad n\mathbf{b}\left(\mathbf{D}_{x,x}\mathbf{L}(x,y),\ldots,\mathbf{L}(x,y)\right),\end{aligned}$$

where d is the degree of $\mathbf{B}$. Similar formulas give the derivatives with respect to y and the mixed partials [Man92]. With this information, we can compute the normal and second fundamental form at a point on $\mathbf{S}$.

The boundaries between S-patches pose a slight problem when approximating a network of S-patches. Since the S-patch surface might be only G^1, the second fundamental forms of two S-patches meeting along a boundary do not in general agree. For the cubic interpolant approximation, two different second fundamental forms will be used, one for each S-patch of the original surface. Note that the resulting approximation will only be G^1 at the vertices.

However, this use of two second fundamental forms introduces a difficulty with the boundary curves of the patches. Consider the situation at the boundary between two S-patches. Two cubic interpolant patches will be constructed, one on each side of the S-patch boundary. If the curvatures used to construct these two boundary curves fails to match, then two different geometric Hermite curves will be constructed, and the approximation will not be C^0. To

	S-Patch Branch		Cubic Interpolant
Sides	4	5	3
Depth	3	6	3
Number in surface	20	4	238
Control points per patch	20	210	10
Total control points	1240		2380
Relative Cost Per Point	2	42	1
Total Cost For Surface	1000		238

Table 2. Approximation of S-patch branch surface.

avoid this problem, I sample the tangents and curvatures directly from the boundary curves of the S-patches. A non-planar geometric Hermite curve is then fit to this data. Note also that if the S-patches are of depth no greater than three, then we can use the boundary of the S-patches as the boundary curves for the cubic interpolant patches.

4.3.1 Results

I give two examples of approximating an S-patch surface with the cubic interpolant. Both S-patch surfaces are from [LD90, Loo92]. The initial mesh for the first example (the branch mesh) appears on the left in Figure 5. Shaded images of the S-patch surface, of the cubic interpolant approximation to this surface, and of the refinement map appear in Figure 9.

The branch S-patch surface is comprised of twenty-four S-patches. Twenty of these are four sided patches of depth three, while four are five sided patches of depth six. The cubic interpolant surface was refined three times; the final approximation is ε-G^1 for $\varepsilon = 0.04$ degrees and has 238 patches.

A comparison of the storage requirements appears in Table 2. With no sharing of vertices, the S-patch surface requires 1240 vertices. The cubic interpolant requires 2380 vertices for the approximation. If sharing is used, the cubic interpolant storage cost decreases to roughly half (1100 points), while the S-patch scheme decreases to about 1000 points.

The primary advantage to using the cubic interpolant to approximate S-patches is in time required to evaluate a point on the surface. The cost to evaluate S-patches of high degree is expensive compared to the cost of evaluating the cubic interpolant. For example, it is about 42 times as expensive to evaluate a depth six, five sided S-patch than it is to evaluate the cubic interpolant. The cost increases dramatically as the number of sides increases; as seen in the next example, it is 92 times as expensive to evaluate a depth six, six sided S-patch than it is to evaluate the cubic interpolant.

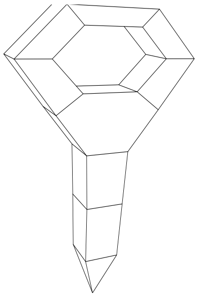

Figure 6. Ring mesh.

The total cost of tessellating the example S-patch surface is about 4.2 times as expensive as evaluating its cubic interpolant approximation. Note that the tessellation of the cubic interpolant approximation has 2.4 times the number of triangles in the S-patch tessellation. Thus, the cost per triangle of the tessellation of the S-patch is about 10 times as expensive as the cost for the cubic interpolant. Further, the S-patch evaluation costs used in the table only account for the evaluation of the Bézier simplex; they do not include the cost of evaluating the **L** function.

The second example is of the ring mesh shown in Figure 6. Shaded images of the initial S-patch surface and the approximation appear in Figure 10. A comparison of the costs and storage is given in Table 3. Here the evaluation cost for the S-patch surface is about 4.2 times that of the cubic interpolant approximation, while the number of triangles in the cubic interpolant tessellation is about 3 times that of the S-patch surface. The cubic interpolant surface is ε-G^1 for $\varepsilon = 0.052$ degrees. (The number of control points in a depth six, six sided S-patch is unrelated to the number of cubic interpolant patches in the approximation; the fact that these two quantities are equal in this example is a coincidence.)

As a final note, S-patches have a large number of control points. Ideally, our approximation of the S-patch surfaces would require less storage. This was not the case in either of the above two examples. However, in these examples, we could not hope to get a reduction in storage cost as the initial cubic interpolant approximations have roughly the same number of control points as the S-patch surface (1000 and 1480 for the cubic interpolant surfaces versus 1240 and 1768 for the S-patch surfaces).

This large number of control points needed in the initial approximation is due to approximating the bicubic patches with cubic interpolant patches (20 control points for the bicubic patch versus 80 control points for the four cubic interpolant patches). The cubic interpolant initial approximation to S-patches with more sides results in a significant reduction in control points (462 for a six sided, depth six

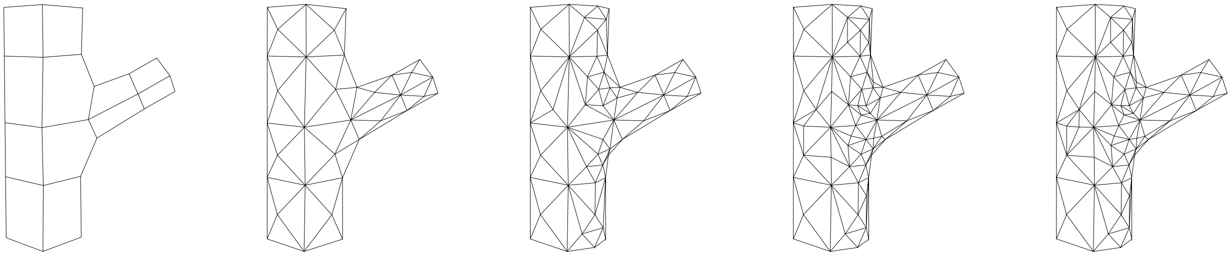

Figure 5. The branch mesh and its refinements.

	S-patch Ring			Cubic Interpolant
Sides	3	4	6	3
Depth	6	3	6	3
Number in surface	4	31	2	462
Control points per patch	56	20	462	10
Total control points	1768			4620
Relative Cost Per Point	5.6	2	92	1
Total Cost For Surface	1379			462

Table 3. Approximation of S-patch ring surface.

S-patch versus 60 for six cubic interpolant patches). Thus, a reduction in storage costs could be achieved by approximating only the high degree S-patches, leaving the bicubic patches in tensor product form.

5. Conclusions

In this paper, I have presented a construction for a triangular polynomial surface patch that interpolates the data at three points to second order. I then used this patch to make ε-G^1 approximations of algebraic surfaces, offset surfaces, and S-patches.

There are several areas for further research relating to the cubic interpolant and approximate continuity. First, the bounding of the discontinuity in surfaces normals along the boundaries is quite simplistic. For cubic patches, the discontinuity can be determined analytically by finding the roots of two polynomials, one of eighth degree, the other of nineteenth degree. However, better techniques should be devised.

The original motivating problem was to fit a smooth surface to parametric scattered where only positional information is known. Ideally, we could use the cubic interpolant to interpolate such data. Extending the cubic interpolant to approximate such data sets would require estimating normals and second fundamental forms at the data points, and possibly generating additional data points if refinement is necessary.

References

[CC78] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *CAD*, 10:350–355, 1978.

[CG91] G. Celniker and D. Gossard. Deformable curve and surface finte elements for free-form shape design. In *Proceedings of SIGGRAPH 1991*, pages 257–265, July 1991.

[dBHS87] C. de Boor, K. Höllig, and M. Sabin. High accuracy geometric Hermite interpolation. *CAGD*, 4(4):269–278, December 1987.

[DLG90] Nira Dyn, David Levin, and John A. Gregory. A butterfly subdivision scheme for surface interpolation with tensi on control. *ACM Transactions on Graphics*, 9(2):160–169, April 1990.

[DM92] T. DeRose and S. Mann. An approximately G^1 cubic surface interpolant. In T. Lyche and L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design II*, pages 185–196. Academic Press, 1992.

[HDD$^+$92] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of SIGGRAPH 1992*, pages 71–79, July 1992.

[LD89] C. Loop and T. DeRose. A multisided generalization of Bézier surfaces. *TOG*, 8(3):204–234, July 1989.

[LD90] C. Loop and T. DeRose. Generalized B-splines of arbitrary topology. In *Proceedings of SIGGRAPH 1990*, pages 347–356, 1990.

[LMD92] M. Lounsbery, S. Mann, and T. DeRose. Parametric surface interpolation. *Computer Graphics and Applications*, 12(5):45–52, September 1992.

[Loo87] Charles Teorell Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, 1987.

[Loo92] C. Loop. *Generalized B-spline Surfaces of Arbitrary Topological Type*. PhD thesis, Univerity of Washington, Seattle, WA, 1992.

[Man92] Stephen Mann. *Surface Approximation Using Geometric Hermite Patches*. PhD thesis, University of Washington, Seattle, December 1992.

[MLL$^+$92] S. Mann, C. Loop, M. Lounsbery, D. Meyers, J. Painter, T. DeRose, and K. Sloan. A survey of parametric scattered data fitting using triangular interpolants. In H. Hagen, editor, *Curve and Surface Design*, chapter 8, pages 145–172. SIAM, 1992.

[MS92] H. Moreton and C. Séquin. Functional optimization for fair surface design. In *Proceedings of SIGGRAPH 1992*, July 1992.

[Ram87] Lyle Ramshaw. Blossoming: A connect-the-dots approach to splines. Technical Report 19, Digital Systems Research Center, Palo Alto, Ca, 1987.

[SD] J. Schweitzer and T. DeRose. Applying blossoming to algebraic curves and surfaces. In preparation.

[War98] Joe Warren. Subdivision surfaces course. Technical report, SIGGRAPH 2000 Course on Subdivision Surfaces, 1998.

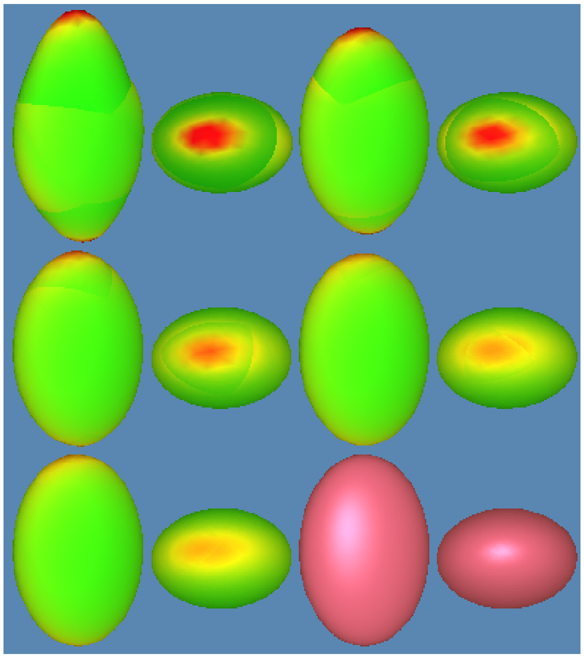

Figure 7. Curvature plots of surfaces fit to $4x^2 + 2y^2 + z^2 - 4 = 0$. The bottom row shows the curvature plot and the surface after four refinements. (The final surface is ε-G^1 for $\varepsilon = 0.4$ degrees.)

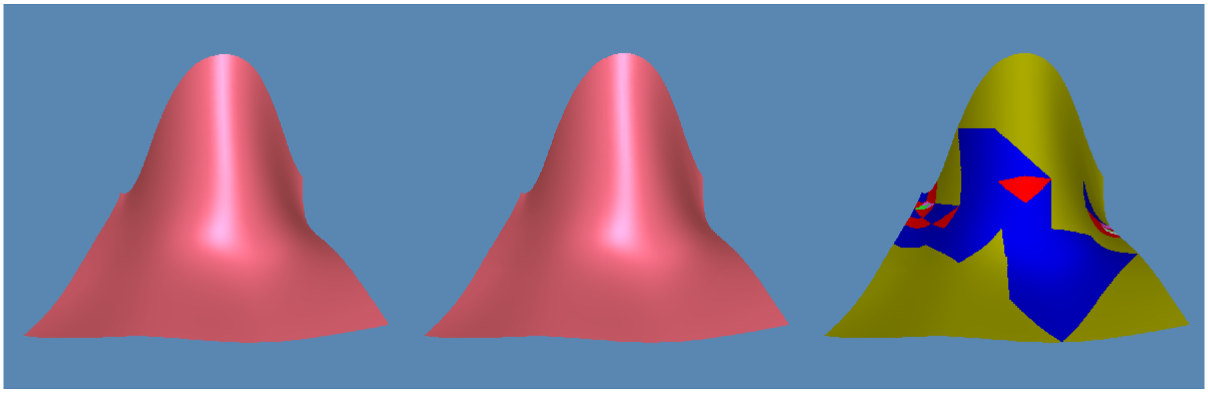

Figure 8. An offset to a bicubic.

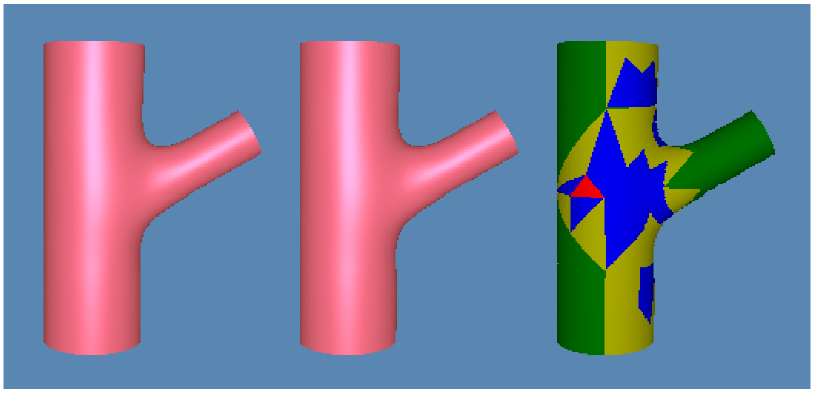

Figure 9. The S-patch “branch” surface.

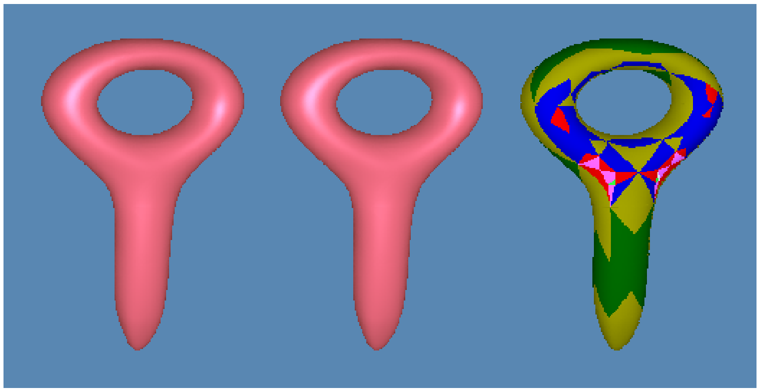

Figure 10. The S-patch “ring” surface.